

# **NASA Ames En route Simulation (RTO58, Phase 3)**

Document Number: FAA-AP-2001-0792

August 27, 2001

STATUS: Final

Provided to NASA Ames in Response to Contract: NAS2-98004

Lockheed Martin Air Traffic Management  
DSR Program  
9211 Corporate Boulevard  
Rockville, MD 20850

---

## Preface

NASA Ames has funded Lockheed Martin Air Traffic Management (LMATM) under Task Order #58 to perform engineering analysis and trade study necessary for the recommendation of a Display System Replacement (DSR) simulation system that supports Center-TRACON Automation System (CTAS) tool development and Distributed Air-Ground Traffic Management research.

The purpose of the cost or estimation data that is provided in this paper is to assist NASA Ames in the evaluation of the alternatives and sizing possible future efforts. This cost or estimation data does not constitute a proposal by LMATM to perform the stated work.

# Table of Contents

<b>1.</b>	<b>PURPOSE .....</b>	<b>1</b>
<b>1</b>	<b>INFORMAL GUIDANCE.....</b>	<b>2</b>
1.1	PHASE 2 CONCLUSION SUMMARY .....	3
1.2	NASA AMES SIMULATOR REQUIREMENTS.....	5
1.3	NASA AMES SIMULATOR DERIVED REQUIREMENTS.....	6
<b>2</b>	<b>NASA AMES SYSTEM BASELINE.....</b>	<b>8</b>
2.1	AIRSPACE OPERATIONS LAB .....	8
2.2	CTAS LABORATORY .....	12
2.2.1	CTAS Subsystems.....	12
2.2.2	Direct-To.....	15
<b>3</b>	<b>DSR SIMULATOR OPTIONS .....</b>	<b>16</b>
3.1	SIMULATION VERSUS STIMULATED OPTIONS.....	17
3.1.1	Stimulation Option: Stimulated DSR and Host.....	18
3.1.1.1	Overview.....	18
3.1.1.2	Detailed Design.....	21
3.1.1.3	Technical Issues.....	22
3.1.1.4	Summary of Advantages and Disadvantages.....	22
3.1.1.5	Hardware Configuration and Cost.....	23
3.1.2	Simulation Option: Simulated DSR and Host.....	24
3.1.2.1	Overview.....	24
3.1.2.2	Detailed Design.....	26
3.1.2.3	Technical Issues.....	28
3.1.2.4	Summary of Advantages and Disadvantages.....	28
3.1.2.5	Hardware Configuration and Cost.....	29
3.1.3	Hybrid Option: Simulated DSR and Stimulated Host.....	30
3.1.3.1	Overview.....	30
3.1.3.2	Detailed Design.....	33
3.1.3.3	Technical Issues.....	35
3.1.3.4	Summary of Advantages and Disadvantages.....	35
3.1.3.5	Hardware Configuration and Cost.....	36
3.1.4	Option Recommendation.....	37
3.2	CONSOLE SIMULATION.....	39
3.2.1	Console Frame.....	39
3.2.1.1	Technical Issues.....	40
3.2.1.2	Summary of Advantages and Disadvantages.....	40
3.2.1.3	Recommendation.....	40
3.2.2	Keyboard and Trackball.....	40
3.2.2.1	Technical Issues.....	4241
3.2.2.2	Summary of Advantages and Disadvantages.....	4241
3.2.2.3	Recommendation.....	4241
3.2.3	Cost.....	4241
3.3	NASA AMES TO DSR SIMULATOR INTERFACE .....	4342
<b>4</b>	<b>CONCLUDING RECOMMENDATIONS .....</b>	<b>4443</b>
4.1	SYSTEM SUMMARY .....	4443
4.2	DSR SIMULATOR USE .....	4746
4.3	FUTURE WORK.....	5150
<b>10</b>	<b>APPENDIX A: HOST S/390 EMULATION.....</b>	<b>5352</b>

10.1	OVERVIEW .....	<del>53</del> 52
10.2	SYSTEM ARCHITECTURE .....	<del>54</del> 53
10.3	TECHNICAL ISSUES .....	<del>56</del> 55
10.4	COST .....	<del>57</del> 56
<b>20</b>	<b>APPENDIX B: DSR SIMULATION.....</b>	<b><del>58</del>57</b>
20.1	R-POSITION CONSOLE SIMULATION.....	<del>59</del> 58
20.2	D-POSITION CONSOLE SIMULATION.....	<del>63</del> 61
<b>30.</b>	<b>APPENDIX C: ACRONYMS.....</b>	<b><del>64</del>62</b>

---

## Figures

FIGURE 1 NASA AMES SYSTEM LOGICAL ARCHITECTURE.....	28
FIGURE 2 ADRS ARCHITECTURE.....	10
FIGURE 3 CTAS SYSTEM.....	12
FIGURE 4 DIRECT-TO PHYSICAL IMPLEMENTATION.....	15
FIGURE 5 STIMULATION FUNCTIONAL ARCHITECTURE.....	19
FIGURE 6 STIMULATION LOGICAL ARCHITECTURE.....	20
FIGURE 7 STIMULATION PHYSICAL IMPLEMENTATION.....	21
FIGURE 8 SIMULATION FUNCTIONAL ARCHITECTURE.....	24
FIGURE 9 SIMULATION LOGICAL ARCHITECTURE.....	25
FIGURE 10 SIMULATION PHYSICAL IMPLEMENTATION.....	26
FIGURE 11 HYBRID FUNCTIONAL ARCHITECTURE.....	3130
FIGURE 12 HYBRID LOGICAL ARCHITECTURE.....	32
FIGURE 13 HYBRID PHYSICAL IMPLEMENTATION.....	33
FIGURE 14 DSR CONSOLE FRAME.....	4039
FIGURE 15 DSR KEYBOARD AND TRACKBALL.....	4241
FIGURE 16 NASA AMES TO DSR/HOST SIMULATOR.....	4342
FIGURE 17 HOST DIRECT STIMULATION OF DSR AND ADRS.....	4746
FIGURE 18 CONTROLLER STIMULATION OF HOST.....	4847
FIGURE 19 ADRS STIMULATION OF HOST WITH RADAR DATA.....	4947
FIGURE 20 ADRS STIMULATION OF DSR WITH NEW MESSAGE.....	4948
FIGURE 21 CONTROLLER STIMULATION OF ADRS / HADDS WITH NEW MESSAGE.....	5049
FIGURE 22 HADDS HOST INTERACTION.....	5049
FIGURE 23 HOST S/390 EMULATOR WITH PERIPHERALS.....	5554
FIGURE 24 HOST RADAR INPUT LOGICAL THREAD.....	5655

---

## Tables

TABLE 1 SUMMARY OF SITE EVALUATIONS.....	3
TABLE 2 SUMMARY OF DSR FIDELITY AND PRIMARY LAB USAGE.....	4
TABLE 3 NASA AMES INITIAL ASSESSMENT.....	6
TABLE 4 DSR SIMULATION PRODUCT COST.....	29
TABLE 5 DSR SIMULATOR AND HOST STIMULATOR PRODUCT COSTS.....	36
TABLE 6 REQUIRED CAPABILITIES FOR DSR SIMULATOR OPTIONS.....	37
TABLE 7 DESIRED CAPABILITIES FOR DSR OPTIONS.....	38
TABLE 8 DSR CONSOLE AND PERIPHERAL EQUIPMENT.....	4341
TABLE 9 TOTAL HARDWARE PROCUREMENT FOR THREE OPTIONS.....	4544
TABLE 10 SOFTWARE DEVELOPMENT COST RANGE SUMMARY.....	4544
TABLE 11 FUTURE SOFTWARE DEVELOPMENT WORK.....	4544
TABLE 12 DSR STANDALONE TRAINING SYSTEM AND NASA AMES DSR SIMULATOR DIFFERENCES.....	5655
TABLE 13 HOST STIMULATOR PRODUCT COST.....	5756
TABLE 14 EXISTING DSR SIMULATOR FUNCTIONS.....	6159
TABLE 15 REQUIRED UPGRADES TO DSR SIMULATOR.....	6160
TABLE 16 EXCLUDED FUNCTIONS FROM DSR SIMULATOR.....	6361
TABLE 17 D-POSITION SIMULATION.....	6361

---

# 1. Purpose

This report represents the completion of Phase 3 of the en route simulation fidelity options for Advanced Air Transportation Technologies Task Order RTO-58. LMATM in Rockville, Maryland is performing this task order for NASA Ames Research Center at Moffett Field, California. This task order is composed of 3 phases.

During Phase 1 of the task order, LMATM provided NASA Ames a description of the Air Route Traffic Control Center (ARTCC)-sector environment and LMATM learned about the simulation needs at NASA Ames.

During Phase 2 of the task order, LMATM provided a high level comparison of the features and capabilities of the major ARTCC-sector simulation facilities. Evaluation of each site was not comprehensive, rather the site's capability, approach, and limitations in emulating key elements identified in Phase 1 were evaluated.

During the final phase of the task order, Phase 3, LMATM will present a set of options and make a recommendation for the simulation of the ARTCC-sector environment at NASA Ames based on the information gathered from Phase 2 of the task order.

---

# **1 Informal Guidance**

Since the task order provided from NASA Ames does not provide a formal set of requirements that the proposed DSR simulator must meet, this section specifies the set of informal requirements for the NASA Ames DSR simulator that both NASA Ames and LMATM have agreed upon.

The requirements in this section are based on Phase 2 conclusions, the task order itself, and information supplied by NASA Ames.

## 1.1 Phase 2 Conclusion Summary

The table below summarizes laboratory capabilities, as reported for each of the ARTCC simulation sites that were evaluated. The sites that were evaluated were:

- Technology Insertion Lab (TIL) at LMATM.
- Mid-America Aviation Resource Consortium (MARC).
- NASA Ames Research Center.
- Center for Advanced Aviation System Development (CAASD) at Mitre.
- Human Factors Laboratory (HFL) at the W. J. Hughes Technical Center.
- DSR System Support Complex (DSSC) at the W. J. Hughes Technical Center.
- Integration and Interoperability Facility (I2F) at the W. J. Hughes Technical Center.
- Embry Riddle Aeronautical University (ERAU).

Some scores within the table are qualified by plus and minus signs. For example, NASA Ames, the CAASD, and the HFL all have a medium hardware fidelity. However, Phase 2 of the study has shown that the HFL has the highest hardware fidelity, followed by CAASD, then by NASA Ames.

Criteria	TIL	MARC	NASA Ames	CAASD	HFL	DSSC	I2F	ERAU
Hardware Fidelity	High	High	Med -	Med	Med +	High +	High +	Low
Software Fidelity	High	High	Low	Med	Med	High	High	Low
Software Emulation	Yes	No	No	Yes	Yes	No	No	No
Maximum Stress Capacity	No	No	No	No	No	Yes	No	No
Use of Commercial Off the Shelf (COTS) products	Low	Low	High	Med	Med	Low	Low	Med
Cost	High	High	Med	Med	Med	High +	High	Low
End of Life Issues	High	High	Low	Low	Low	High	High	Low

*Table 1 Summary of Site Evaluations*

The study found that there are relationships between the use of the lab and the fidelity found at the lab. The three purposes of the labs that were evaluated are development and test, research and development and training. The relationships are illustrated in [Table 2](#).

Site	High Fidelity	Medium Fidelity	Low Fidelity	Primary Lab Usage
I2F	X			Development and Test
DSSC	X			Development and Test
TIL	X	X	X	Development and Test
HFL		X	X	Research and Development
NASA Ames			X	Research and Development
Mitre		X	X	Research and Development
MARC	X			Training
ERAU			X	Training with some research

*Table 2 Summary of DSR Fidelity and Primary Lab Usage*

For sites that have a primary use of development and test, the site needs multiple layers of fidelity (high, medium and low) to satisfy their simulation needs. The high fidelity systems include the actual replication of the system that must be maintained as the system evolves. Accordingly, there has to be a plan in place to fund the upgrading of the system in order for the fidelity of the system to stay high. Also, the use of COTS products help to reduce the end of life issues present at a lab by making the products less expensive to replace.

The research and development providers also provided multiple levels of system fidelity, ranging from medium to low fidelity. These sites were more focused on the evolution of the system and demonstration of new functionality rather than replicating the system. Accordingly, there was not the need for maintaining the cost of a high fidelity system.

The training systems provided the fidelity necessary for the aims of the training program. For example the simulation fidelity of the MARC system is high. This high fidelity allows graduates from the MARC site to go directly to the centers for air traffic control, and not have to go to the FAA Aeronautical Academy. However, the low fidelity system found at the ERAU does not provide the same level of controller experience as the MARC system such that the ERAU graduates must go to the FAA Aeronautical Academy to continue their controller training.

In addition it was observed that a layered approach to simulation is most successful for the development of new ATC functionality:

- Low level fidelity is best for rapid concept development. Examples of these tools are Visual Basic or PowerPoint.
- Medium level fidelity is necessary for the evaluation of function by users. This level of fidelity is high enough for the users to interact with the proposed system and develop opinions on the systems strengths and weaknesses. Examples of these tools are high level X based software development tools, such as Orthogon ODS Toolbox, Gallium InterMAPics and Eagan Viewman.
- High level fidelity is necessary for test and evaluation of the prototype under operational conditions. The high fidelity requires substantial portions of the operational system to be replicated.

## ***1.2 NASA Ames Simulator Requirements***

The task order proposal supplied by NASA Ames to LMATM states that the NASA Ames simulator must have:

- One to three sector positions within one ARTCC area/specialty.
- The required NASA Ames laboratory environment identified in Phase 1. This is described in detail in Section 2.
- The desirable characteristics and capabilities found at the evaluated sector-ARTCC sites during Phase 2 of this task order.

### 1.3 NASA Ames Simulator Derived Requirements

During Phase 2, based on a review of NASA Ames simulator needs and similar Research and Development sites around the United States, it was recommended that the following updates be made to the NASA Ames simulation capabilities in the following key areas, as identified in [Table 3](#)[Table 3](#)[Table 3](#).

Criteria	Assessment, Current → Future	Current Assessment & Future Recommendation Rationale
Hardware Fidelity	Medium – → Medium +	<u>Current</u> : While there are several Sony Main Display Monitors (MDM) enclosed in Standard Terminal Automation Replacement System (STARS) consoles, there are no DSR consoles, flight strip printers, or DSR peripheral equipment such as DSR keyboards or trackballs. <u>Future</u> : Better DSR console mockups with DSR keyboards.
Software Fidelity	Low → Medium	<u>Current</u> : NASA Ames has low level en route simulation capability. Track data are generated by simulators such as the Pseudo Pilot Simulator and are transmitted to the Plan View Graphic User Interface subsystem for display. However, NASA Ames does not have a simulator or operational code for the DSR or Host systems. <u>Future</u> : Either simulate Host and DSR or utilize the fielded software.
Maximum Stress Capacity	None	<u>Current</u> : NASA Ames has no capacity and no need for maximum stress DSR testing. <u>Future</u> : No Change.
End of Life Issues	Low → Medium	<u>Current</u> : NASA Ames uses mostly COTS hardware and custom built software, so there are few end of life issues. <u>Future</u> : With increased fidelity, some end of life DSR equipment may need to be procured.

Table [333](#) NASA Ames Initial Assessment

Based on findings in [Table 3](#)[Table 3](#)[Table 3](#) and other information, the set of derived requirements for the NASA Ames DSR simulator is:

- Positions: Both DSR radar and data positions are required. The DSR simulator must support most R-position and D-position commands.
- Data Consistency/Routing: Data that is output by the DSR simulator must also be transmitted to the NASA Ames subsystems for update of the NASA Ames subsystem clients. In addition, some NASA Ames unique messages from R and D-positions must be routed to the NASA Ames systems.
- Center Adaptation: Simulation of various centers, such as Dallas Ft. Worth, Denver and Albuquerque are required. This requires access to Host adaptation data for the centers.
- Radar Data: NASA Ames currently receives Host output from Dallas Ft. Worth. The DSR simulator is required to display live and recorded radar data.

The following capabilities of the DSR simulator are desired by NASA Ames:

- Flight Strips: NASA Ames desires the generation and printing of paper flight strips at the D-position.
- Scenarios: NASA Ames desires to have greater capability of scenario generation than the Pseudo Aircraft Simulator system offers.
- Weather and Radar Processor (WARP): The display of WARP data is desired. This requires a separate WARP gateway processor and the WARP Display Application that runs in the R-position console. This topic was not investigated for the DSR simulator at this time.  
Note: The same weather products can be acquired from the Internet.

The following requirements are not required by NASA Ames and so are not addressed in this proposal:

- Multi-user simulation: NASA Ames does not require the DSR simulator to support multiple concurrent DSR simulations. The DSR simulator will run one scenario at a time.
- Enhanced Direct Access Radar Channel (EDARC): The EDARC system is not required for the DSR simulator.
- User Request Evaluation Tool (URET) Core Capability Limited Deployment (CCLD): URET CCLD displays and input commands are not required.
- Ghost Pilots: NASA Ames expects to continue to use the Pseudo Aircraft Simulator for the control and simulation of aircraft at ghost pilot positions.

- Voice Switching Communications System (VSCS): The existing VSCS simulator system employed at NASA Ames will be used for air ground communications for the DSR simulator positions. NASA Ames currently records the VSCS data.

## 2 NASA Ames System Baseline

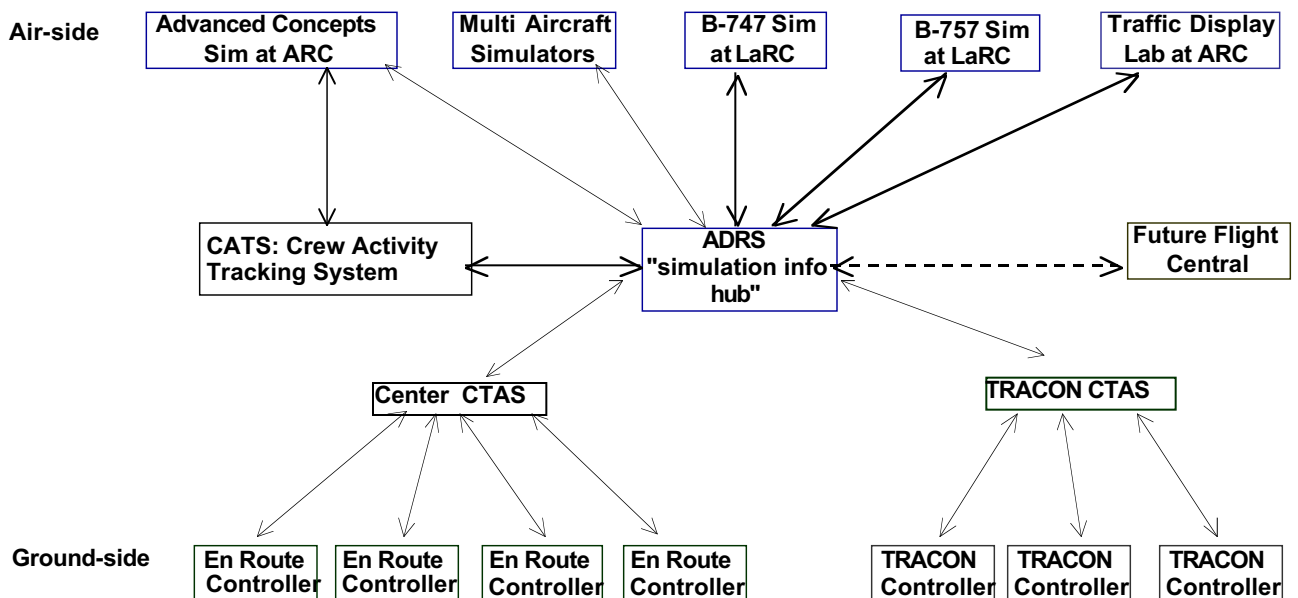
This section reviews the existing NASA Ames system baselines in order to provide the reader a basis for understanding recommendations made later in the report. Information for this section was gathered from visits to NASA Ames, conversations with NASA Ames staff and documents obtained from NASA Ames.

Two laboratory systems were reviewed:

- AOL (AOL).
- CTAS Laboratory.

### 2.1 Airspace Operations Lab

As illustrated in [Figure 1](#), the NASA Ames simulation system in the AOL is composed of air-side systems and ground-side systems. An air-side system's function, such as the Multi Aircraft Simulator or the Pseudo Aircraft System (PAS), is to control the aircraft and also generates position information (latitude, longitude and altitude) for aircraft. Aircraft and position data are sent to the Aeronautical Datalink and Radar Simulator (ADRS) subsystem. The integration of Future Flight Central function



with ADRS is currently only planned, as indicated by the dotted line.

*Figure 1 NASA Ames System Logical Architecture*

A surveillance function within ADRS simulates multiple Center and TRACON radars. ADRS uses 12 radars in the ZFW (Dallas Fort Worth) area that is specified in an ADRS database. ADRS initially selects the radar site closest to the radar return of the aircraft. That radar remains selected until the aircraft nears its maximum range from the radar and ADRS then selects new radar for that aircraft. ADRS does not perform mosaicing of radar.

ADRS corrects the aircraft position and then an alpha beta tracker function determines ground speed and heading of the track. Simulated radar data (range and azimuth) are available to clients as perfect source data, noisy raw data, predicted data and alpha beta tracked data.

ADRS calculates the sector number of the position that has track control only when CTAS is connected. The sector number calculation is based on the location of the aircraft (center coordinates [X, Y] and altitude) and the Host sector boundaries specified in the CTAS adaptation. This may not match Host sector track control that is dependent on controller handoffs.

The surveillance, trajectory and flight data is stored by ADRS. When requested, ADRS forwards this data to the connected Center or TRACON Input Source Manager (ISM) at an update rate specified in the configuration file or a command line option. ADRS can provide range, azimuth, position and beacon code as well as most flight data that is available to Host. In the case that an aircraft is simulated to have Automatic Dependent Surveillance equipment, the ADRS forwards the Automatic Dependent Surveillance data to CTAS at a selected update rate.

When client applications request the data, ADRS sends the data and updates to that data to the client. ~~Figure 2~~~~Figure 2~~ shows ADRS distributing the data to the two ISM CTAS clients. The ISM manages communication and data processing for the many sources of flight plan and track data that CTAS may receive, and passes any two-way messages back to the relevant data sources via the appropriate daemon. The purpose of ISM is to transform disparate data from multiple sources into source-independent messages.

Whenever an ISM connects to ADRS, ADRS immediately forwards all of its valid flight plans that it received from its connected sources (e.g. Pseudo Aircraft Simulator and flight simulators). In order for the ISM to connect to ADRS, the ISM follows a communication protocol with ADRS:

- ISM establishes the socket connection with ADRS.
- The ISM and ADRS handshake.

- ADRS sends aircraft flight plan to the ISM.
- ADRS sends aircraft state data for radar simulation to the ISM.
- ISM sends requests for traffic data for display to ADRS.
- ADRS sends periodic traffic data to ISM.
- ADRS sends data link messages to the ISM.

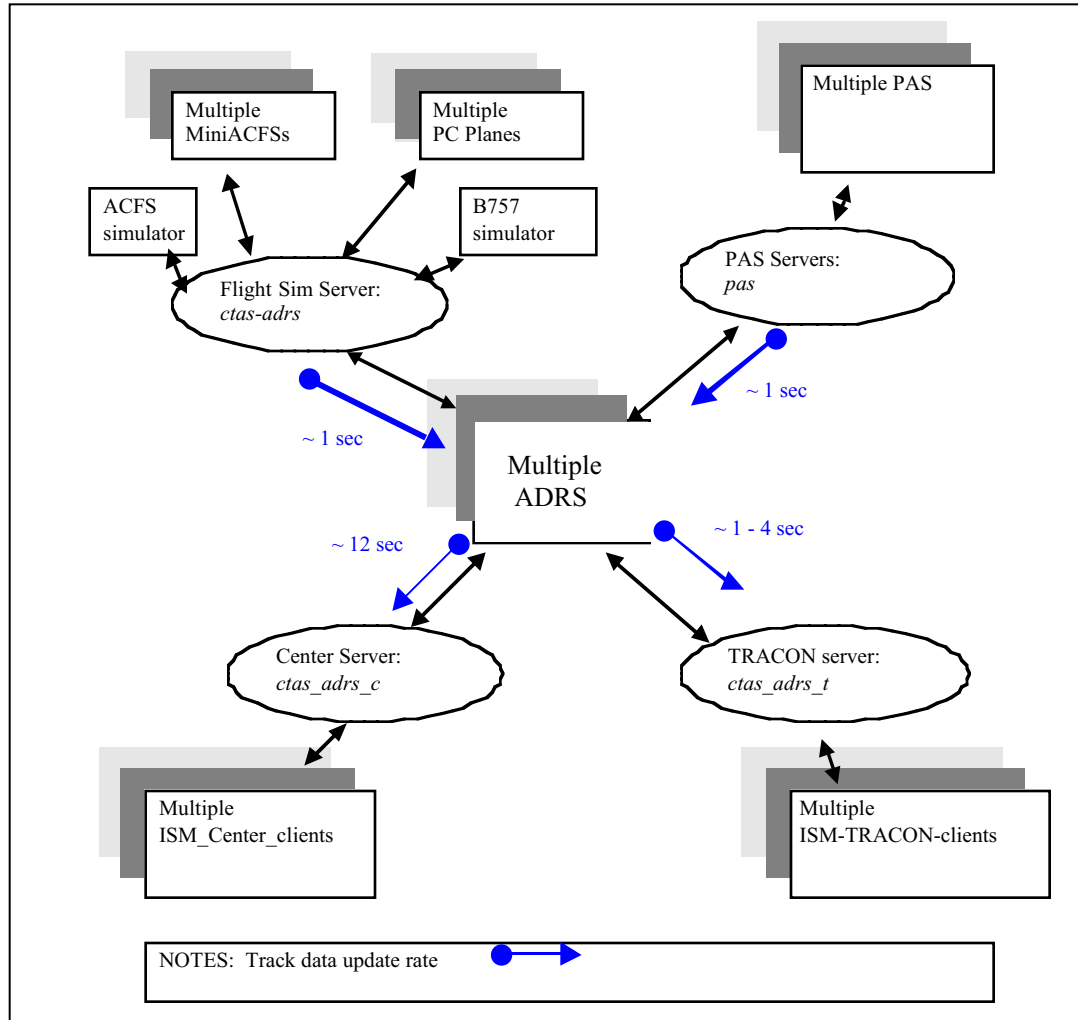


Figure 22-2 ADRS Architecture

The CTAS system, composed of tools such as Traffic Management Advisor, Decent Advisor and others, processes the data. CTAS transmits the result to the Plan Graphics User Interface (PGUI) for display. The AOL periodically installs a copy of the CTAS system from the CTAS laboratory for research purposes.

The PGUI software displays radar and flight data at either en route or TRACON controller positions. The boxes labeled “En Route Controller” and “TRACON Controller” in [Figure 1](#) [Figure 1](#) [Figure 1](#) represent the PGUI functionality. Typically, the PGUI simulates an entire center, but can also be configured to simulate a single sector. The PGUI provides a graphical user interface for the display of data such as flight plans, tracks, targets, limited data blocks, estimated time of arrivals, scheduled time of arrivals, descent advisories, timelines, conflicts, direct-to information and graphical weather information. The PGUI also supports the entry of most of the typical flight parameters, such as route amendments, assigned altitudes, temporary altitudes, and handoffs.

While the PGUI does not provide the look and feel of the DSR system, it can be used for initial prototype, engineering and demonstration purposes of the en route system when a realistic environment is not necessary. Thus, a DSR simulator does not replace the PGUI subsystem, rather it augments the PGUI by providing a higher fidelity simulation of the en route system.

## 2.2 CTAS Laboratory

As mentioned in the system level discussion, the CTAS system represented in [Figure 1](#) is a copy of the CTAS system used by the NASA Ames AOL. The CTAS group has a separate laboratory that is used to develop CTAS concepts. Much of this chapter is derived from <http://www.ctas.arc.nasa.gov/>.

This section will review the CTAS subsystems and the Direct-To project. The Direct-To project is interesting because it's goal is to integrate CTAS displays into the actual DSR R-position software.

### 2.2.1 CTAS Subsystems

The set of subsystems that compose CTAS is found in [Figure 3](#). A brief explanation of the relevant subsystems follows the diagram.

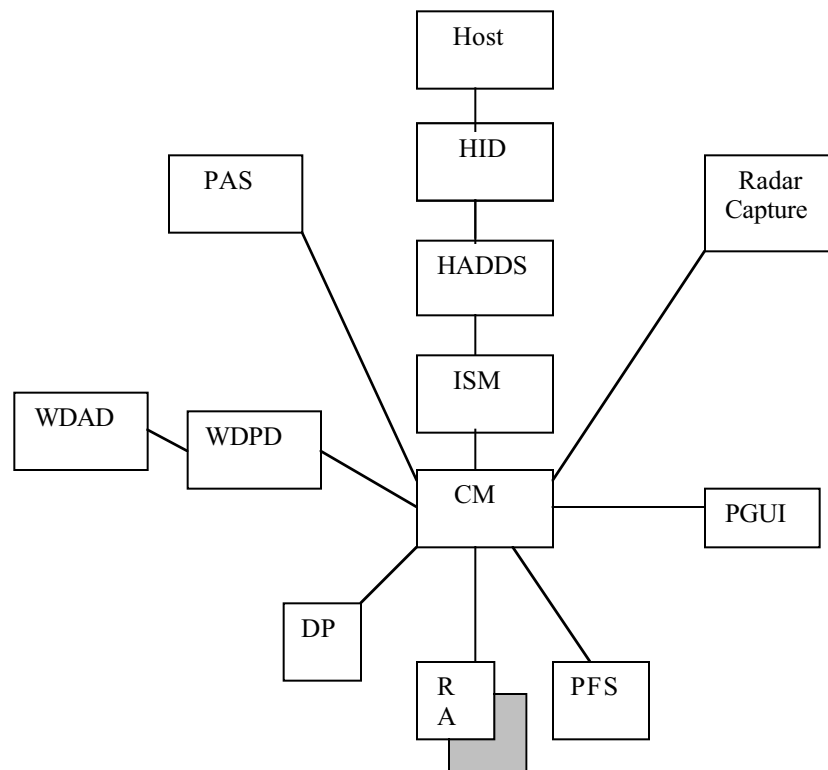


Figure 333 CTAS System

### Host:

The Host provides live radar feed and Common Message Set (CMS) data to the CTAS system.

### Host Interface Devices (HID):

The HID links the Host to the NAS LAN to support communications between Host and Traffic Flow Management applications. This function can also be provided by the Host 320 patch that also supports the transmission of the CMS data.

### Host ATM Data Distribution System (HADDs):

HADDs is a product of Free Flight Phase 1 and interfaces with Host to provide CMS data to systems such as URET, CTAS and Controller Pilot Data Link Communication. For example, CTAS calculates data such as schedule time of arrival and provides that data through the HADDs for display on the DSR R-position display. In addition, CTAS provides flight plan amendments to the Host through the HADDs.

CMS data received from the Host by HADDs and passed to CTAS includes:

- Flight Plans describing the aircraft type, route of flight, cruise altitude, cruise speed, and take-off time for each aircraft.
- The route as it is converted into individual waypoints by the Host
- Radar tracking data giving the position, altitude, and speed of all aircraft at 12 second intervals.
- Amendments to and deletion of flight plans
- Controller commands to CTAS such as manual swap, aircraft resequence, and suspend/unsuspend messages

### Input Source Manager (ISM):

The ISM has the same functionality as the ISM used in the AOL.

### PAS:

Pseudo Aircraft Systems (PAS) is a computerized flight dynamics and piloting system designed to provide a high fidelity, multi-aircraft, real-time simulation environment to support air traffic control research. PAS outputs surveillance data in latitude longitude form rather than system coordinates.

### Communications Manager (CM):

The CM manages the communications among all of the CTAS processes shown connected to it in the figure above. CM receives aircraft tracks and flight plans sent by the ISM. CM can also directly process captured radar data. CM uses the aircraft data to

create and maintain its own aircraft database. CM informs clients when an aircraft has been added to or deleted from CTAS.

CM is the intermediary for any message passed from one client process to another, and all messages pass through CM on their way to their intended destination. CM manages the connection and initialization of all the client processes. Once connected, CM completes their initialization, including sending all current aircraft and airspace configuration information of interest in the form of messages.

The CM graphics user interface permits the user to do the following:

- Request the display of aircraft information such as aircraft type, track status (tracked, coasting, planned), estimated time of arrival status, or flight plan.
- Dynamically connect to and disconnect from all valid flight plan and track data sources.
- Start and stop data recording or playback.
- Initiate two-way communications with the Host.
- Initiate use of live weather.
- Configure the ground speed and vertical speed filters.

#### Weather Data Acquisition Daemon (WDAD):

The WDAD process is a perl script that is responsible for gathering weather data files and making them available on the CTAS network file system. The current format of the input data is known as AWIPS211 and comes from the NOAA Rapid-Update Cycle computational process.

#### Weather Data Processing Daemon (WDPD):

The WDPD is responsible for converting raw weather files provided via the WDAD into binary weather files usable by CTAS. Currently, WDPD processes incoming AWIPS211 files, which are provided in the form of a Lambert conformal grid that covers the continental United States. The grid data includes wind component speeds, temperature, and pressure versus latitude/longitude.

#### PGUI:

The PGUI has the same functionality as the PGUI used in the AOL.

Information about the CTAS tools listed below may be acquired from the CTAS web site.

- Dynamic Planner (DP)
- Route Analyzer (RA)
- Profile Selector (PFS)

### 2.2.2 Direct-To

NASA Ames Task Order 39 is currently being implemented at LMATM Rockville. This project incorporates a user interface at the R-position for the CTAS Direct-To system. The Host function is emulated by a product called NAS In A Box. The Host outputs CMS data that passes through the HID to the HADDS. The HADDS system transmits the data to the CTAS functions. CTAS generates products such as the Direct-To Advisory View, Sector Conflicts View and Direct-To route graphics. These data are transmitted through a Display System Gateway (DSGW) and are displayed at the R-position.

This functional flow is illustrated in [Figure 4](#).

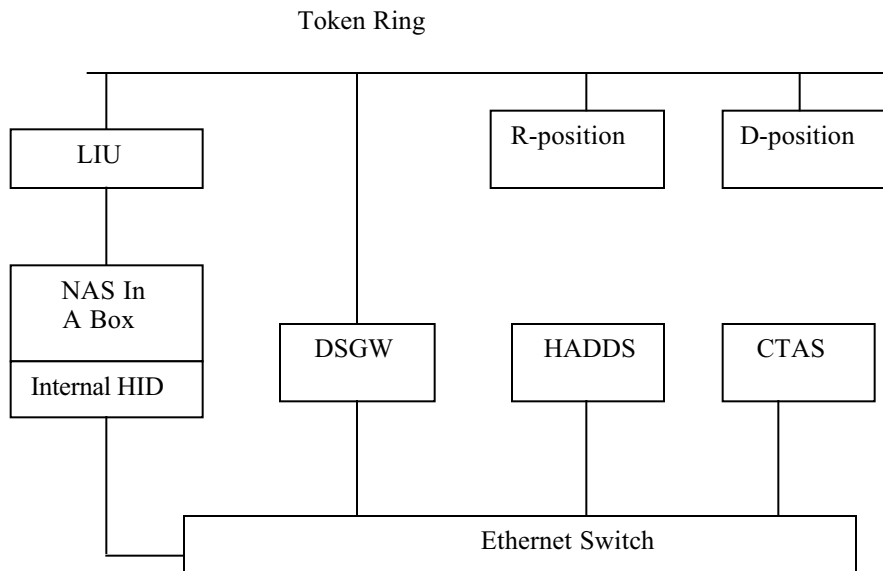


Figure 4 Direct-To Physical Implementation

---

### 3 DSR Simulator Options

This section analyzes the options available for a DSR simulator and is subdivided into two sections:

1. Stimulation versus Simulation Options: This section evaluates the benefits of stimulation of the DSR system versus a simulation of the DSR system. Refer to Section 3.1.
2. DSR Console Hardware: This section identifies the portions of the DSR console hardware should be procured, such as DSR keyboards, trackballs, consoles (frames) and flight strip printers. Refer to Section 3.2.

A set of options is presented and analyzed for each topic. A recommendation is then made for each option at the end of each topic. In addition, a concluding recommendation is made at the end of this report that integrates each topic recommendation into a single solution.

### **3.1 Simulation versus Stimulated Options**

In context of this paper, a simulator is defined as a software application that provides a subset of the functions that the operational air traffic control software (Host and DSR) provides. The larger the number of supported functions by the simulator, then the higher fidelity of the simulation.

On the other hand, a stimulator is a software application that provides simulated external inputs to a fielded copy of the operational Host and DSR software.

Simulator, stimulator and a hybrid of the two approaches are evaluated:

- Stimulation: This option proposes that the NASA Ames DSR simulator utilize both the fielded DSR and Host software. This solution stimulates the actual Host and DSR code. In this option, some of the DSR hardware is utilized.
- Simulation: This option proposes the simulation of both DSR and Host subsystems. Since the solution proposes simulation, much of the fielded DSR and Host hardware would be replaced by commercial off the shelf hardware.
- Hybrid: This option proposes a hybrid of the stimulation and simulation options. This solution proposes that the actual Host software running on a S/390 emulator and simulated DSR software be used. This solution uses mostly commercial off the shelf hardware.

For both the stimulation and hybrid solutions, actual fielded Host code would be executed on an S/390 emulator. The S/390 emulator is part of a DSR Standalone Training System that is currently being developed by LMATM for the FAA. The details of this system are found in Section 10, Appendix A.

### 3.1.1 Stimulation Option: Stimulated DSR and Host

This solution proposes using the actual DSR and Host production code that runs on actual DSR hardware and a Host S/390 emulator respectively.

#### 1.1.1.13.1.1.1 Overview

A functional architecture diagram of the option is shown in ~~Figure 5~~~~Figure 5~~Figure-5. The inputs to the system (CTAS, HADDS, ADRS and Host Direct SIM tape) are shown on the left edge of the diagram. A control interface controls the Host substitute while a user interface controls the R and D-position displays. Both of these interfaces are shown at the bottom edge of the diagram. System output is located on the right edge of the diagram: display data to R and D-positions, ADRS, Flight Strips, and HADDS. Areas in light gray indicate software that requires development. Note that the DIU is only partially located in the gray area indicating that the DIU functionality is mostly complete. The ADRS transmits long range radar data to Host via the Host substitute. Within the Host Substitute Interface (HSI), the Peripheral Adapter Module Replacement Item (PAMRI) function transforms the long range radar data (in common digitizer 2 format) and transmits the radar data to the Host. Note that the Host Substitute can not process the short range radar format (common digitizer ASR).

The Host processes the radar data and produces targets, limited data blocks, and tracks. Host transmits this data to the DIU. The DIU receives this data, determines the processor destination and transmits the data to that sector position(s). That sector position receives the data and displays the data on the MDM. In order to keep the NASA Ames system data in synchrony with the DSR system, the DIU also forwards the track data to the ADRS.

The ADRS transmits flight data to Host via the DIU. New flight plans, modifications to flight plans (e.g. amend, activation) and deletions to flight plans are sent from ADRS to Host, through the DIU. The DIU identifies these messages as interfacility messages to Host. The Host processes the flight data and transmits the appropriate flight data (flight plans) to the DIU.

The DIU receives this data, determines the processor destination and transmits the data. That sector position receives the data and displays the flight plan data on the MDM. In order to keep the NASA Ames system data in synchrony with the DSR system, the DIU also forwards the flight data to the ADRS.

The ADRS also directly provides the DIU flight and radar data for display at the position displays, bypassing the Host.

Data recorded on the Direct SIM can be played back through the Host substitute. The output radar and flight data is forwarded to the DIU. The DIU transmits the data both to the sector position(s) and also to the ADRS.



of the DSR system. ADRS can also transmit data directly to the D and R-positions through the DIU. CTAS sends data to the R-positions via the DSGW (part of the Direct-To program).

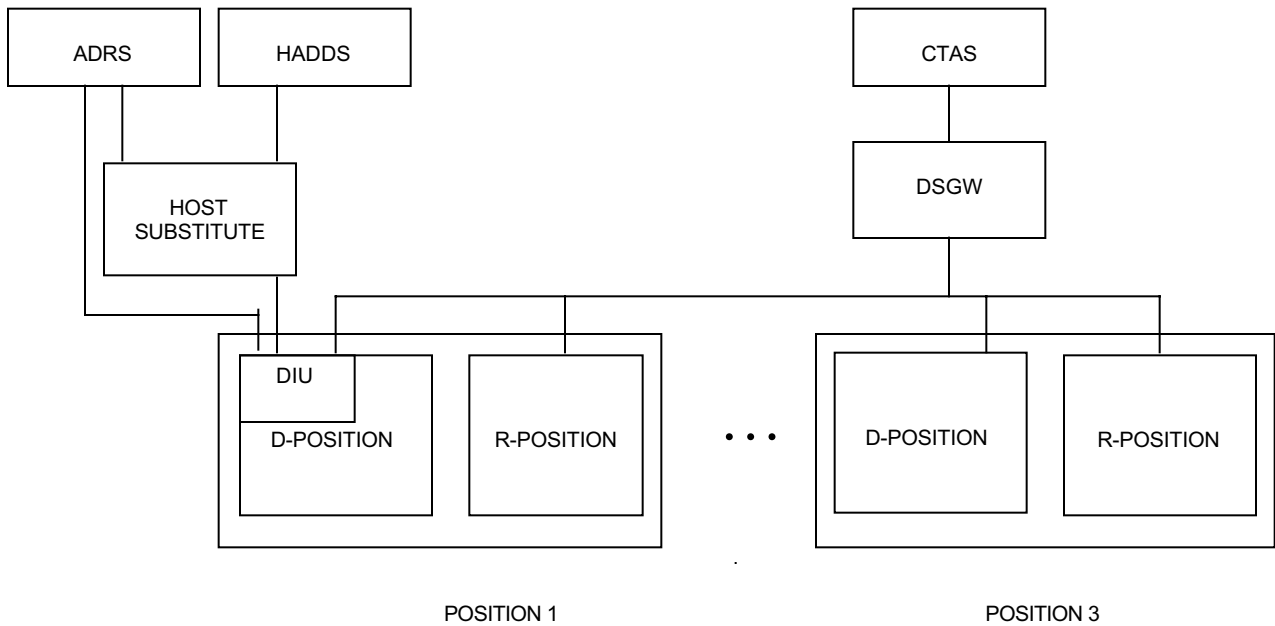


Figure 666 Stimulation Logical Architecture

The physical implementation of the proposed DSR stimulation system and its interface to the NASA Ames subsystems is shown in Figure 7. The Host Substitute and the DIU share the same Ethernet as the NASA Ames systems. The DIU is a gateway that converts channel messages from the Host to messages sent over the token ring to the R and D-positions.

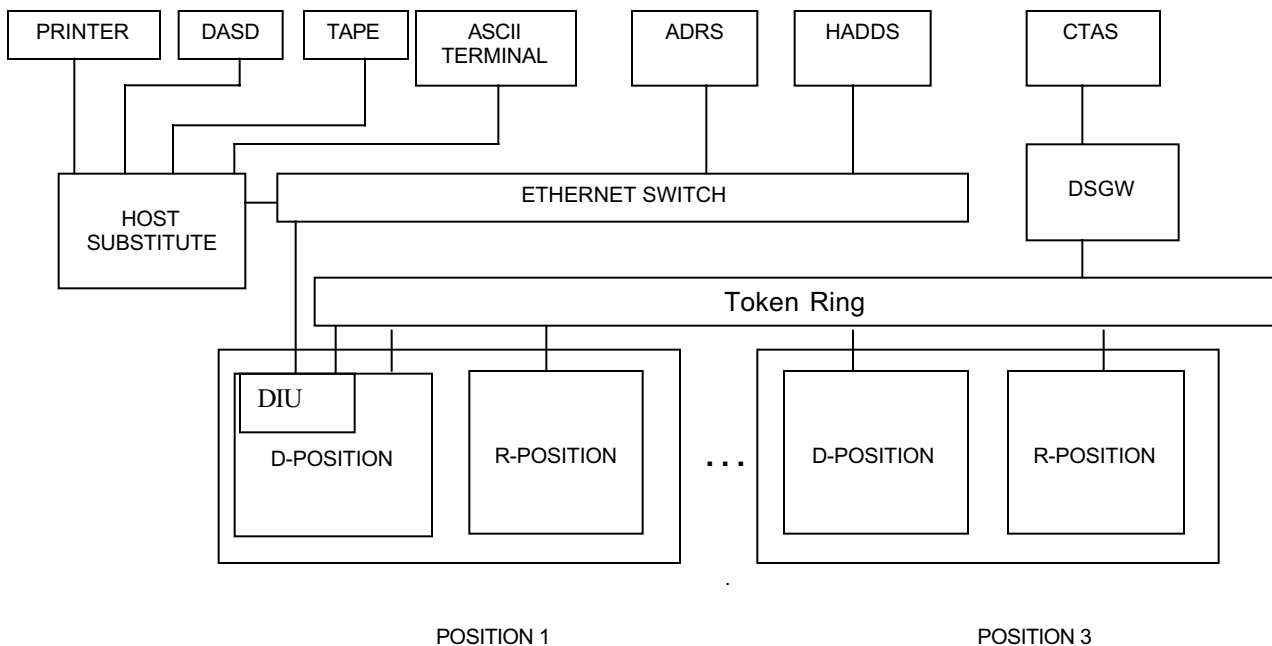


Figure 777 Stimulation Physical Implementation

### 3.1.1.2 Detailed Design

This section provides greater option detail.

#### 4.1.1.1.43.1.1.2.1 DSR and Host Code

This option uses the actual production code for Host and DSR, except for the DIU that has been modified in the following ways:

- The DIU provides a connection to the HSI via an Ethernet. This modification is being made to support the DSR Standalone Training System program.
- The DIU accepts data from the ADRS system to transmit to Host and also potentially transmit to sector positions for display.

#### 3.1.1.2.2 System Control

The System Management CSCI controls the DSR system.

#### 4.1.1.1.33.1.1.2.3 Operating system

The operating system uses AIX 3.2.5.

#### 4.1.1.1.43.1.1.2.4 Display Subsystem

Data is displayed using Remote Graphics Language that is processed through a Raytheon Console Display Generator.

#### **4.1.1.1.53.1.1.2.5 Adaptation**

The DSR system uses both Host and DSR adaptation. The Host adaptation allows the DSR simulator to run various configurations from different ARTCC centers.

#### **4.1.1.1.63.1.1.2.6 Scenario Playback and Generation**

Host Direct Simulation data can be loaded onto the virtual disk within the Host substitute and can drive the DSR system with radar and flight data. System Analysis Recording (SAR) recording can be provided with additional hardware and software development.

#### **4.1.1.1.73.1.1.2.7 Simulation of new functions**

ADRS transmits NASA Ames specific data, such as CTAS data, to the DSR system via the DIU. ADRS uses the adapted Sector Display Number map to identify the destination processor ID to the DIU. The DIU transmits the CTAS data to the appropriate position for display.

In order to display data at the sector position, new DSR messages may need to be developed, and changes to the Host Format Conversion (HFC) code made to receive these messages and integrate the data into the position display. It is possible that modifications to Display Services (DS) will be necessary.

### ***3.1.1.3 Technical Issues***

- None.

### ***3.1.1.4 Summary of Advantages and Disadvantages***

#### **Advantages**

- High software fidelity.
- Future DSR and Host releases are available for upgrade.
- There are several advantages of having a Host system:
  - High fidelity for radar and flight displays.
  - Flight data available for display to control at the R and D-positions.
  - Flight Strip data available to controllers.
  - D-positions can amend flight plans.
  - Dynamic Simulator (DYSIM) and Host Direct Simulation are available for scenario generation.
  - Center Host adaptation data is available to easily simulate various centers.

#### **Disadvantages**

- Difficulty in obtaining the RISC/6000 Model 7018 processors. These are IBM micro-channel machines and not PCI machines.
- Difficulty in obtaining the Raytheon Console Display Generators.
- Modifications to Display Services DSR display code are made in Remote Graphic Language rather than in X.
- Dependency on AIX 3.2.5 that is at end of life and now maintained by LMATM. The current DSR system runs AIX 3.2.5.
- Maintenance of NAS In A Box system:
  - Ongoing expense to maintain the Host system.
  - Must reinstall new releases of DSR and Host code periodically to remain current.
  - Difficult to make modifications to support research tasks.
- Prototype modifications to the DSR R and D-position software may need to be reincorporated when new DSR code level is released.
- Requires extensive support environment to maintain Host and DSR code.

#### ***3.1.1.5 Hardware Configuration and Cost***

The hardware configuration and cost for the NAS In a Box is located in [Table 13](#) ~~Table 13~~ on page [57](#) ~~56~~.

In addition to this, is the cost of the RISC/6000 7018 and the Console Display Generator, both that are at end of life and not able to be commercially procured. Thus, the prices of these items are not available.

There is also a medium sized software development cost to modify the DIU, as detailed above. The DIU subsystem would have to be merged each time that NASA Ames upgrades to a new DSR release.

### 3.1.2 Simulation Option: Simulated DSR and Host

This solution proposes using the simulated DSR and Host code that runs on commercial off the shelf hardware. This approach relies on the NASA Ames subsystems to provide flight and radar data to the DSR system for display since there is no Host.

#### 1.1.1.13.1.2.1 Overview

A functional architecture diagram of the option is shown in [Figure 8](#)~~Figure 8~~[Figure 8](#). The inputs to the system (CTAS and ADRS) are shown on the left side of the diagram. A control interface controls the Host substitute while a user interface controls the R and D-position displays. Both of these interfaces are shown at the bottom edge of the figure. System output is located on the right side of the diagram: display data to R and D-positions, CTAS, and ADRS. Areas in light gray indicate software that is not yet complete. Note that the R-position is only partially in the gray area indicating the R-position simulation is mostly complete.

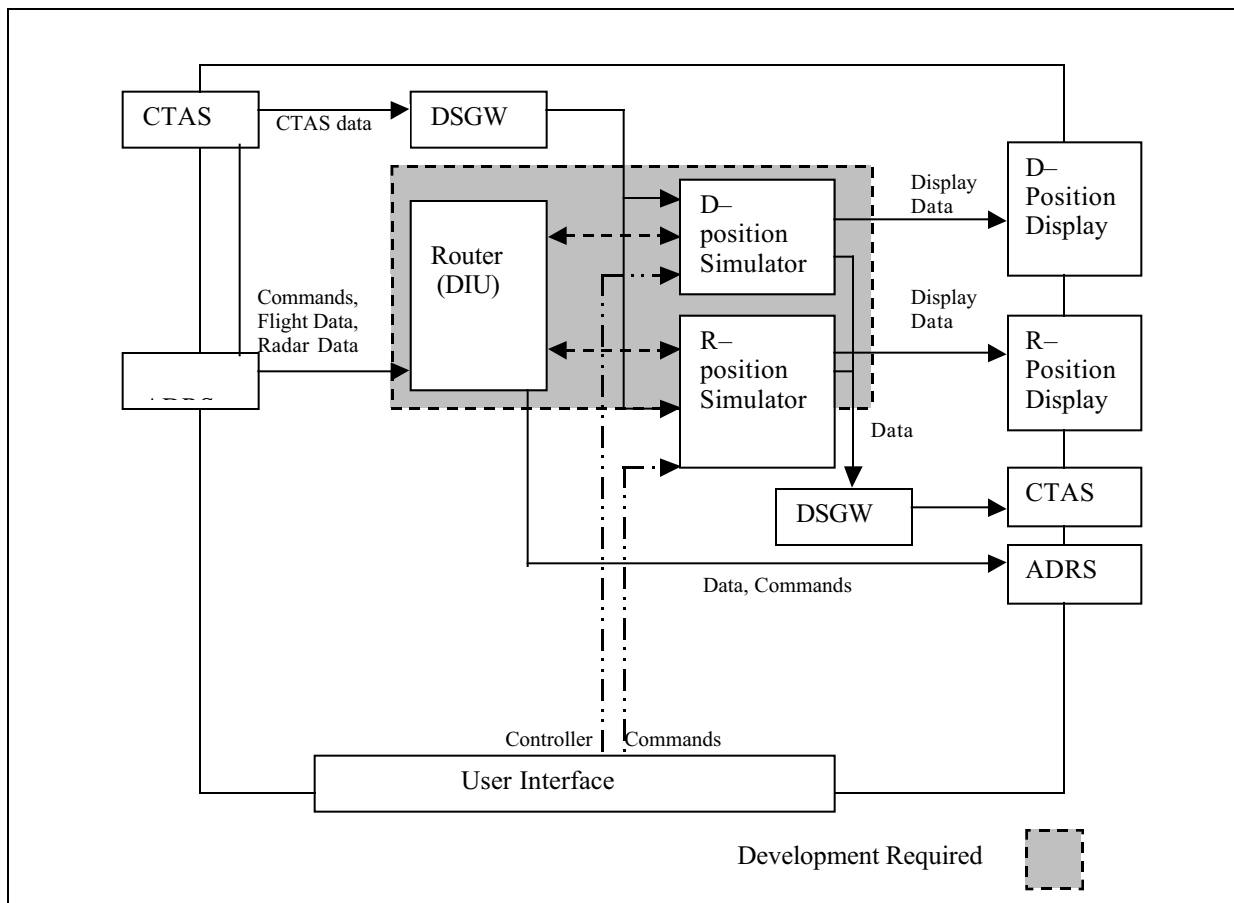


Figure ~~888~~ Simulation Functional Architecture

A major attribute of this option is that there is no Host substitute. Consequently, NASA Ames systems must behave as Host to the DSR simulator. This means that for medium software fidelity, ADRS must provide the DSR simulator the same set of radar and flight data as the Host. The ADRS transmits flight data, such as new flight plans, modifications to flight plans (e.g. amend, activation) and deletions to flight plans, to the DSR simulator via the DIU. The DIU receives this data and transmits the data to the proper sector position. That sector position receives the data and displays the flight plan data on the MDM.

The ADRS also provides the DIU radar data (track, limited data block, targets) for display at the position displays.

The CTAS transmits data to the DSR simulator via the DSGW. The DSGW transmits the data to the proper sector. Responses to the CTAS system are returned to CTAS via the DSGW. CTAS is also able to exchange data with ADRS.

Since there is no Host, there is no Direct SIM capability, no HADDS interface and no paper flight strip generation. The logical architecture of this solution is illustrated in [Figure 9](#). ADRS provides both radar and flight data to the DSR system. This information is received by the DIU and transmitted to the sector positions for display.

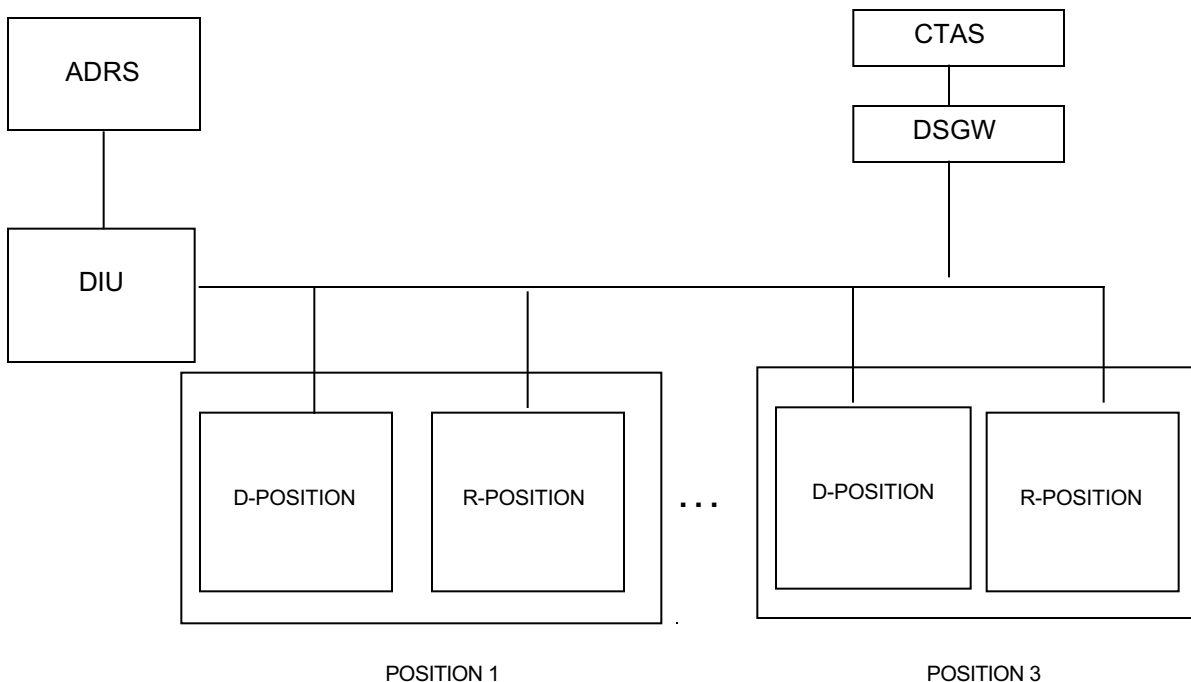


Figure 9 Simulation Logical Architecture

The physical implementation of the proposed DSR simulation system and its interface to the NASA Ames subsystems is shown in [Figure 10](#)~~Figure 10~~[Figure 10](#). The section below the diagram discusses each feature of the system.

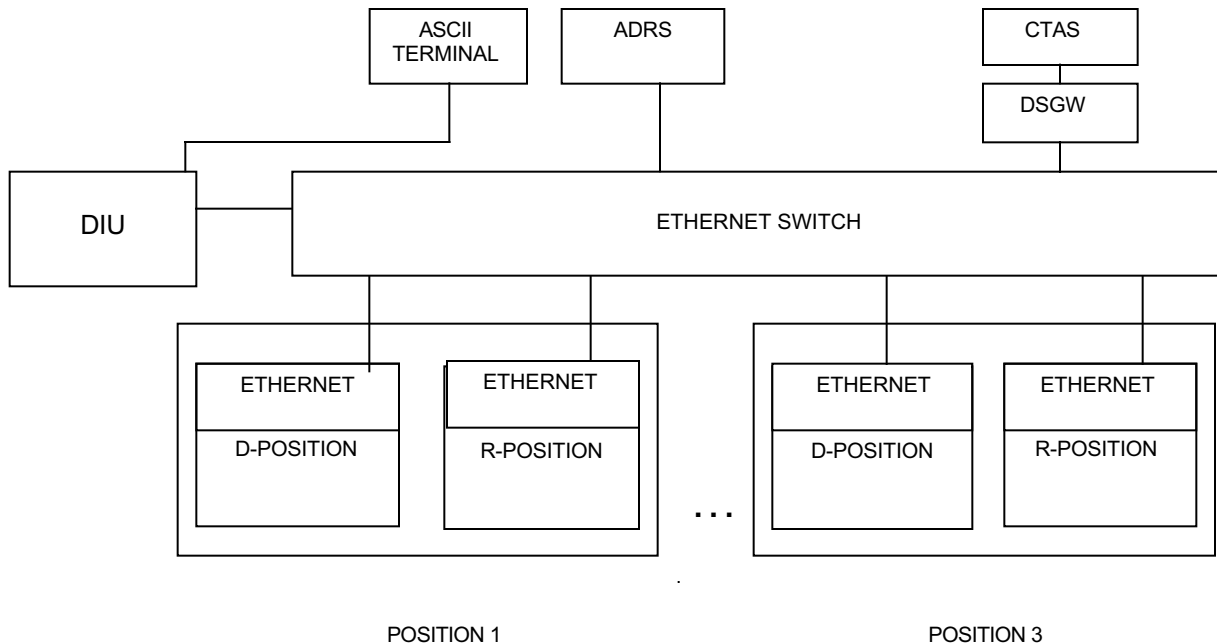


Figure [10](#)~~10~~[10](#) Simulation Physical Implementation

### 3.1.2.2 Detailed Design

This section provides greater option detail.

#### [4.1.1.1.13](#) 1.2.2.1 DSR and Host Code

This option uses simulated code for DSR that is described in section 20, page [57](#)~~56~~[56](#), but uses no Host code.

In addition, since there is no Host, all of the radar and flight data is sent from ADRS or HADDS to DSR. Thus, the NASA Ames systems are simulating Host. The interface between the DSR and ADRS is implemented with a new module, the DIU. The DIU provides the following functions:

- Mapping of sector and position to TCP/IP address: ADRS will address messages sent to the DIU by sector and position. The DIU maps this information to the TCP/IP address and then transmits the message to that address across the Ethernet.
- DIU/ADRS communication protocol: There is a communication protocol expected by ADRS and its clients.

- Processes DSR Messages: The DIU supports the reception of control and data messages from ADRS. A subset of these messages are found in the “DSR/Host Interface Control Document” and are supported by ADRS:
  - Add Target Position Symbols: This message contains target data for display.
  - Clear All Display Data: This message resets the DSR simulation.
  - NAS Text Message Output: This message contains text data to be displayed on the Computer Readout Display view at the DSR R and D-positions.
  - Process Limited Data Block: This message provides Limited Data Block data for display at the DSR R-positions.
  - Process Track Data Block Group: This message provides track data block information for display at the DSR R-positions.
  - Sector Position/Display Number: This message contains sector specific data that DIU transmits to each of the sector processors. This message contains data such as the Display Center X and Y System Coordinates and the Sector Map name that identifies the geomap to display at the sector.
  - Update ATC Starting Status: This message that provides current system time.
  - Other messages as required.
- Health Checks: DIU provides any health check information to the ADRS system.

#### **3.1.2.2.2 System Control**

There is no system control system such as System Management CSCI. Instead, the system is distributed through the network and then started using shell scripts. UNIX sockets are used as the mechanisms for message transmission between processors.

#### **3.1.2.2.3 Operating system**

The operating system uses Sun Solaris, version 2.8. The DSR simulator has minimal operating system dependencies and can easily be updated.

#### **3.1.2.2.4 Display Subsystem**

Data is displayed using X Windows that is processed through a graphics adapter card.

#### **3.1.2.2.5 Adaptation**

Host adaptation files, such as map files, are supplied to DSR.

#### **3.1.2.2.6 Scenario Playback and Generation**

SAR recording can be provided with additional hardware and additional software development. There is no Host Direct Simulation capability since there is no Host.

#### **3.1.2.2.7 Simulation of new functions**

ADRS transmits NASA Ames specific data to the DSR system via the DIU. Integration of new functions is made in X in the DSR simulator code. This solution has X Toolkit capability support.

#### **3.1.2.3 Technical Issues**

- None.

#### **3.1.2.4 Summary of Advantages and Disadvantages**

##### Advantages

- No dependency on end of life technology:
  - RISC/6000 model 7018.
  - AIX 3.2.5.
  - CDG.
  - Token Ring.
- DSR simulation written in standard X that may allow quicker integration with NASA Ames prototype functionality.

##### Disadvantages

- A Host/DSR simulator will incur the costs for development and maintenance (problem resolution and enhancement). This is an additional effort and cost, and may result in periods where the simulator system is not current with the operational system.
- The result is medium simulation fidelity of DSR.
- The result may be low simulation fidelity of Host.
- The following Host functions are not available:
  - Flight Strip data available to controllers.
  - Center adaptation data to easily simulate various centers.
- There is no Host and HID interface to attach HADDS.

### 3.1.2.5 Hardware Configuration and Cost

The table below illustrates the hardware cost for the DSR simulator hardware. These are list prices and should be taken as a rough order of magnitude cost.

<b>Description</b>	<b>Quantity</b>	<b>Unit Price</b>	<b>Extended Price</b>
<b>DSR Simulator Hardware (R-position)</b>			
Server UE10/440, 512MB/20GB*	3	\$4,300.00	\$12,900
Raptor 2000 24M Graphic Accelerator (includes X-Server)**	3	\$10,000.00	\$30,000
10/100Base T F/W UltraSCSI PCI Adapter*	3	\$800.00	\$2,400
17" 1280x1024 Display Monitor*	2	\$400.00	\$800
Solaris 8 Standard, latest release English-only media kit*	1	\$100.00	\$100
Forte C 6 update 2, 1-RTU, Slim Kit for SPARC*	2	\$1,300.00	\$2,600
Subtotal			\$48,800
<b>DSR Simulator Hardware (D-position)</b>			
Server UE10/440, 256MB/20GB*	3	\$3,300.00	\$9,900
10/100Base T F/W UltraSCSI PCI Adapter*	3	\$800.00	\$2,400
17" 1280x1024 Display Monitor*	3	\$400.00	\$1,200
Subtotal			\$13,500
<b>Total Product Cost (less maintenance costs)</b>			\$62,300
* list price from website      ** List price over telephone			

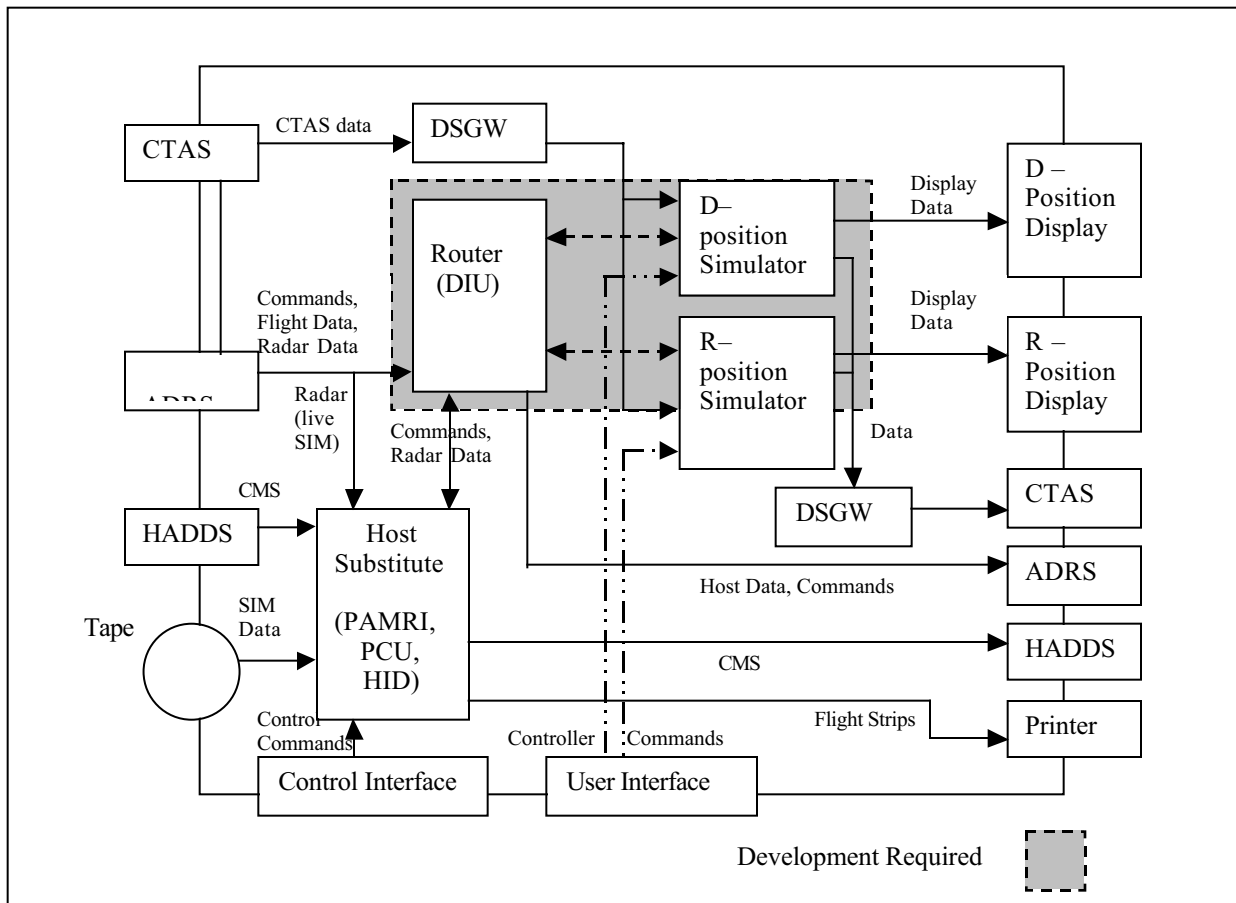
*Table 4 DSR Simulation Product Cost*

### 3.1.3 Hybrid Option: Simulated DSR and Stimulated Host

This option features a simulated DSR system and stimulated Host running on a NAS In a Box.

#### 1.1.1.13.1.1 Overview

A functional architecture diagram of the option is shown in ~~Figure 11~~~~Figure 11~~~~Figure 11~~. The inputs to the system (CTAS, HADDs, ADRS and SIM tape) are shown on the left side of the diagram. A control interface controls the Host substitute while a user interface controls the R and D-position displays. Both of these interfaces are at the bottom edge of the figure. System output is located on the right side of the diagram: Display data to R and D-positions, CTAS, ADRS, HADDs, and a flight strip printer. Areas in light gray (D-position, R-position and DIU) indicate software that requires further software development. Note that the R-position is only partially in the gray area



indicating the R-position simulation is mostly complete.

Figure ~~1111~~ Hybrid Functional Architecture

The ADRS transmits long range radar data to Host via the Host substitute. Within the Host Substitute Interface (HSI), the Peripheral Adapter Module Replacement Item (PAMRI) function transforms the long range radar data (in common digitizer 2 format) and transmits the radar data to the Host. Note that the Host Substitute can not process the short range radar format (common digitizer ASR).

The Host processes the radar data and produces targets, limited data blocks, and tracks. Host transmits this data to the DIU. The DIU receives this data, determines the processor destination and transmits the data to that sector position(s). The D and R-positions are simulated. That sector position receives the data and displays the data on the MDM. In order to keep the NASA Ames system data in synchrony with the DSR system, the DIU also forwards the track data to the ADRS.

The ADRS transmits flight data to Host via the DIU. New flight plans, modifications to flight plans (e.g. amend, activation) and deletions to flight plans are sent from ADRS to Host, through the DIU. The DIU identifies these messages as interfacility messages to Host. The Host processes the flight data and transmits the appropriate flight data (flight plans) to the DIU.

The DIU receives this data, determines the processor destination and transmits the data. That sector position receives the data and displays the flight plan data on the MDM. In order to keep the NASA Ames system data in synchrony with the DSR system, the DIU also forwards the flight data to the ADRS.

The ADRS also directly provides the DIU flight and radar data for display at the position displays, bypassing the Host.

Data recorded on the Direct SIM can be played back through the Host substitute. The output radar and flight data is forwarded to the DIU. The DIU transmits the data both to the sector position(s) and also to the ADRS.

The HADDS transmits and receives CMS data directly from the Host since the Host substitute has an internal HID.

The CTAS transmits data to the DSR simulator via the DSGW. The DSGW transmits the data to the proper sector. Responses to the CTAS system are returned to CTAS via the DSGW. CTAS is also able to exchange data with ADRS.

Flight strips are generated by the Host and are sent to the Host Substitute Interface. There, the strips are transmitted to the flight strip printers over a serial interface.

The logical architecture of this option is illustrated in ~~Figure 12~~~~Figure-12~~~~Figure-12~~. The data sources of the R and D-position displays are:

1. Host within the Host Substitute provides all of the radar or flight data by running Host Direct Simulation data and transmitting the resultant output to the sector positions through the DIU.
2. ADRS provides all of the radar (azimuth range form) and flight data directly to Host. Host processes the data and then transmits the data to the sector positions for display through the DIU.
3. ADRS provides radar (system coordinate form) and flight data directly to the DIU and is displayed at the sector positions. In this case Host is started and connects to the DIU but may not generate any display data.
4. HADDs transmits data generated by CTAS to the substitute Host. In addition, Host transmits CMS data to HADDs for processing by CTAS.

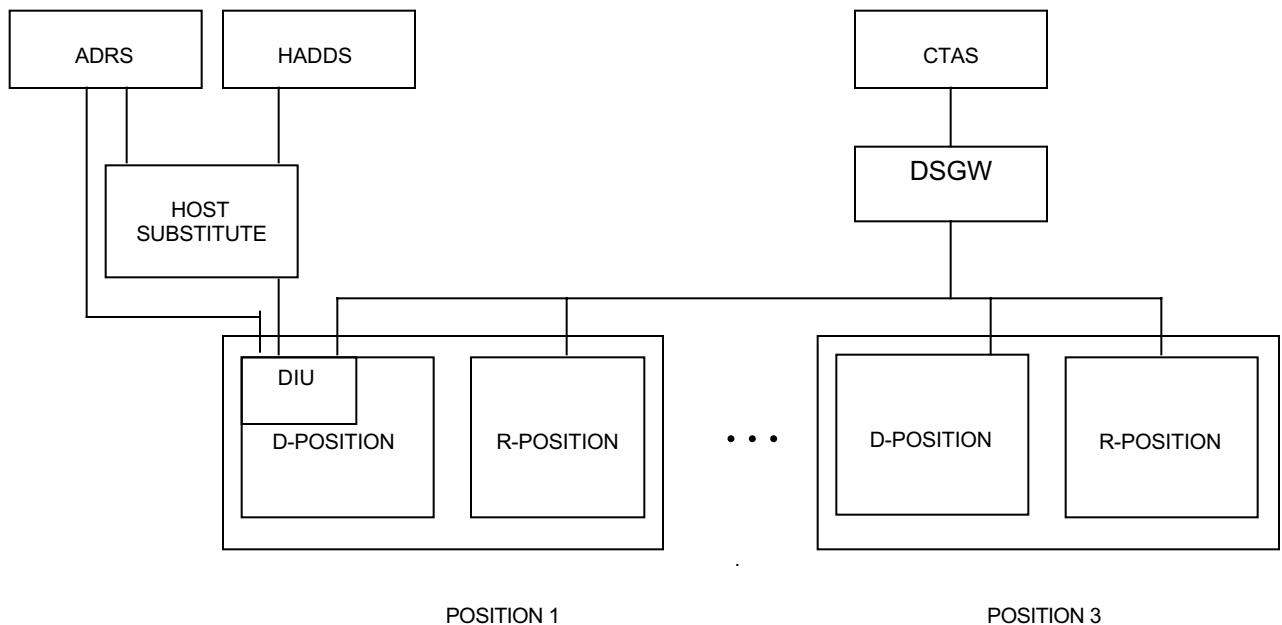


Figure 121212 Hybrid Logical Architecture

The physical implementation of the proposed DSR hybrid system and its interface to the NASA Ames subsystems is shown in Figure 13. The Host Substitute, the DIU and the positions share the same Ethernet as the NASA Ames systems.

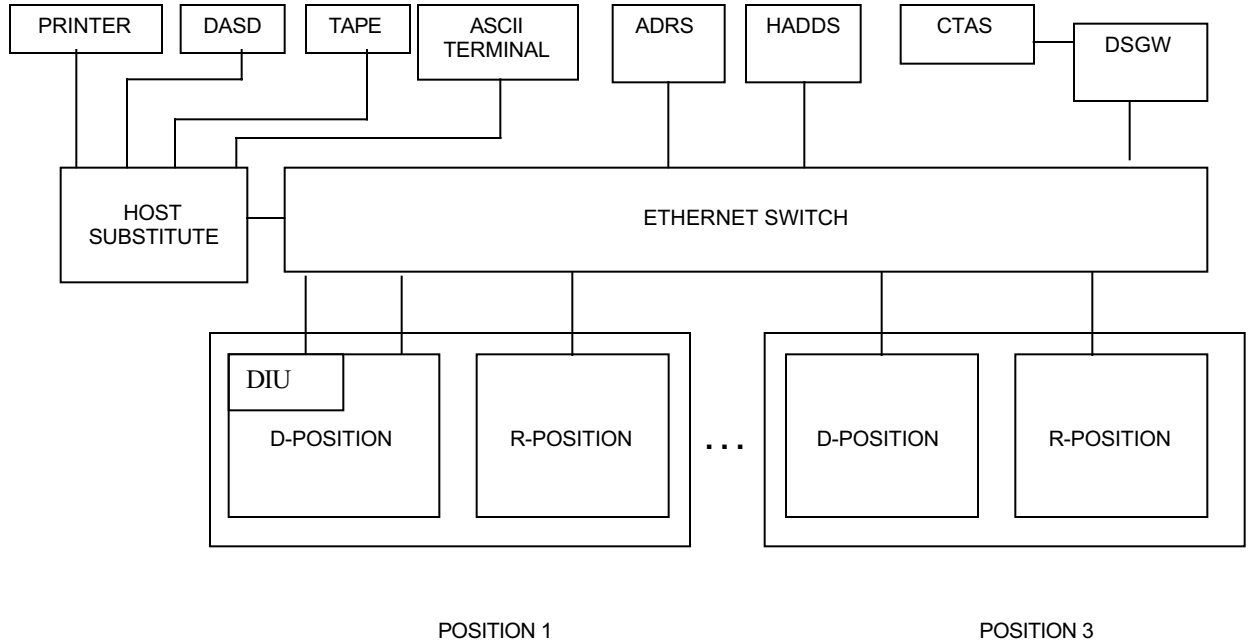


Figure ~~1313~~13 Hybrid Physical Implementation

### 3.1.3.2 Detailed Design

This section provides greater option detail.

#### ~~4.1.4.1.13~~4.1.3.2.1 DSR and Host Code

This option uses Host emulation (refer to section Appendix A: Host S/390 Emulation on page ~~5352~~), simulated code for DSR (refer to section 20, page ~~5857~~), and a modified DIU.

The necessary modifications to the DIU are:

- Connection to the HSI via an Ethernet rather than token ring. While it is planned that the DSR Standalone Training System program will make this modification to the DIU, this design is subject to change. This concern is listed as an issue to track in section 4.3 on page ~~5150~~.
- Data that is output from Host, such as track and flight data, not only are routed to the DSR console, but is also routed to ADRS. This keeps the NASA Ames systems in synchrony with Host and DSR.
- Recognize two types of data transmitted from ADRS: ADRS to Host, or ADRS directly to the D and R-positions.
- Recognize two types of controller commands: Direct commands to Host, or direct commands to ADRS. Commands that are normally sent to Host are

sent to Host. For those new commands associated with prototype functions, the DIU routes the message to ADRS.

#### **3.1.3.2.2 System Control**

The FlightDeck Simulator, provided by LMATM, is used.

FlightDeck provides high levels of availability and real time services for distributed high consequence environments, such as Air Traffic Control systems. FlightDeck provides a consistent Application Program Interface in many environments that provide the following types of services:

- Communication: e.g., broadcast, multicast, and connected data transfer.
- Availability: e.g., detection & recovery from software crashes.
- System Time: e.g., supports time of day.
- System Recording and Analysis: e.g., online and offline analysis tools.
- System Management: e.g., pace distribution and centralized monitor and control.
- Network Management: e.g., supports both OSI and TCP/IP protocol stacks.
- Application builders: e.g., event services.

FlightDeck is currently used for the UK NATS New EnRoute Centre program and the 'UK NATS' New Scottish Centre program, and URET CCLD.

The FlightDeck Simulator was designed to provide an easy to use and repeatable environment that greatly simplifies running over the complete FlightDeck infrastructure. The Simulator uses the exact same FlightDeck API so there are no code differences in the user applications.

Because the simulator is just a library of bound-in routines it cannot provide the full-function distributed environment with centralized management. Instead each executable is started by command (or by script) and management of the running elements in the system is provided by command line tools. Thus, in order to start up several sector processors, shell scripts are written and executed.

#### **3.1.3.2.3 Operating system**

The operating system uses Sun Solaris, version 2.8.

#### **3.1.3.2.4 Display Subsystem**

Data is displayed using X Windows that is processed through a graphics adapter card.

#### **3.1.3.2.5 Adaptation**

The DSR system uses both Host and DSR adaptation. The Host adaptation allows the DSR simulator to easily run various configurations from different ARTCC centers.

#### **3.1.3.2.6 Scenario Playback and Generation**

Host Direct Simulation data can be loaded onto the virtual disk within the Host substitute and can drive the DSR system with radar and flight data. SAR recording can be provided with additional hardware and additional software development.

#### **3.1.3.2.7 Simulation of new functions**

ADRS transmits NASA Ames specific data, such as CTAS data, to the DSR system via the DIU. Integration of new functions is made in X in the DSR simulator code. This solution has X toolkit capability support.

#### **3.1.3.3 Technical Issues**

- None

#### **3.1.3.4 Summary of Advantages and Disadvantages**

##### **Advantages**

- No dependency on end of life technology
  - RISC/6000 7018.
  - Raytheon CDG.
  - Token ring hardware.
  - AIX 3.2.5.
- There are several advantages of having a Host system:
  - High fidelity for radar and flight data
  - Flight data available for display to control at D-position.
  - Flight Strip data available to controllers.
  - D-positions can amend flight plans.
  - Host Direct simulation data can be used as scenario data for driving the system.
  - Center adaptation data is available for other centers to be simulated.
  - There is a Host and HID interface to attach HADDs.

##### **Disadvantages**

- Maintenance of the DSR simulator code:
  - Significant expense to maintain the DSR emulation system.
  - Must augment DSR code to remain current.
- Maintenance of Host:
  - Significant expense to maintain the Host system.
  - Must reinstall new releases of Host code periodically to remain current.

### 3.1.3.5 Hardware Configuration and Cost

The table below illustrates the product cost for the DSR simulator and Host stimulator. This should be taken as a rough order of magnitude cost. This cost does not include the console frames, keyboards, trackballs or flight strip printers. This cost also excludes the DSGW used for a CTAS interface.

Description	Quantity	Unit Price	Extended Price
DSR Simulator*	na	na	\$62,300
NAS Stimulator**	na	na	\$141,200
<b>Total Product Cost (less maintenance costs)</b>			<b>\$203,500</b>

\* The cost of the DSR Simulator hardware is summarized in [Table 4](#)~~Table 4~~[Table 4](#)

\*\* The cost of the NAS Simulator hardware is summarized in [Table 13](#)~~Table 13~~[Table 13](#)

*Table 5 DSR Simulator and Host Stimulator product Costs*

### 3.1.4 Option Recommendation

The tables below summarize the set of required and desired capabilities listed in section 1 that each option meets.

Required Functions	Stimulation Option	Simulation Option	Hybrid Option
D-position Simulation	Yes	Yes*	Yes*
R-position Simulation	Yes	Yes*	Yes*
Data Consistency/Routing	Yes*	Yes*	Yes*
Host Center Adaptation	Yes	No	Yes
Radar Data: Process Live and Recorded	Yes	No	Yes
Hardware Availability	No	Yes	Yes

\* Additional software development necessary

*Table 6 Required Capabilities for DSR Simulator Options*

Data is [Table 6](#)~~Table 6~~[Table 6](#) shows that all three options can provide R and D-positions, and flight-radar commands capabilities. The caveat is that both the simulation and hybrid solutions require further development of the DSR simulator to support these capabilities.

All three options can support some form of data consistency/routing. For the stimulation option to provide data to NASA Ames systems, a gateway between the DSR token ring and the NASA Ames Ethernet is necessary. The simulation option only supports routing of messages between the NASA Ames subsystem and the controller positions since there is no Host. The hybrid option requires some code development in the DIU to route the messages between Host and NASA Ames subsystems, and between console positions and NASA Ames subsystems. Please refer to [Table 10](#)~~Table 10~~[Table 10](#) and [Table 11](#)~~Table 11~~[Table 11](#) on page [45](#)~~44~~[44](#) for the development cost estimates.

All options except simulation support multiple ‘Host center adaptation’ since the simulation option does not provide a Host. Without Host and Host adaptation, aspects of specific centers, such as airspace boundaries, can not easily be simulated.

Since the simulation option does not provide a Host, the option does not allow the DSR system to display Live and Recorded Radar Data. For the stimulation and hybrid solutions, it is planned that the ADRS system provides the Common Digitizer format data to the Host via the DIU.

Desired Functions	Stimulation	Simulation	Hybrid
Flight Strips	Yes	No	Yes
Host SIM Scenario Generation	Yes	No	Yes

*Table 7 Desired Capabilities for DSR Options*

Data in [Table 7](#) shows both the stimulation and hybrid options support the printing of flight strips since both options provide Host emulation. Since there is no Host, the simulation option can not provide strips.

All options except simulation support Host SIM Scenario Generation since the simulation option does not provide a Host. In addition, it is unclear how well the simulator option will support HADDS without the Host or HID interface.

In addition, it is thought that the integration of NASA Ames prototypes with the DSR system would be more feasible if they were integrated into the simulated DSR system rather than the production DSR system. The rationale for this conclusion is that the simulated DSR system utilizes an X – Windows graphics interface for the R–position rather than the Remote Graphics Language interface. The X – Windows system is a more widely used standard than the Remote Graphics Language. Also, the development environment for X is more productive than RGL. For example, the results of X can easily be displayed at normal workstations, rather than at remote MDMs located in a laboratory.

The major obstacles in using the stimulation option is that this option has a dependency on RISC/6000 7018 processors, AIX 3.2.5 and the Raytheon Console Display Generator. All of these products are at end of life and can not be commercially acquired without going to re-marketeers. It is concluded that the equipment and software in the simulation and hybrid options will be more supportable in the future. While a stimulated system provides the highest fidelity, it will incur significant costs in acquisition, up keep, and ease of modification. This dependency effectively rules out the stimulation option.

For the simulation option, the NASA Ames subsystems must simulate Host. While the lack of Host is a significant maintenance simplification, it does prevent NASA Ames from taking advantage of the Host adaptation to easily simulate multiple centers and the Host Direct Simulation capabilities to generate realistic display of radar and flight data. Both of these deficiencies lower the fidelity of the simulation. In addition, without a Host or HID, the HADDS can not interface with the DSR simulator.

For these reasons, it is recommended that NASA Ames select the hybrid option. Note that this option requires significant modifications to the current DSR simulator: incorporation of the Flight Deck simulator infrastructure, updates to the DIU routing capabilities, completion of the interface that transmits messages to Host from the R-position, and other changes. However, this option gives NASA Ames the best flexibility for prototyping functions and sufficient levels of system fidelity.

### **3.2 Console Simulation**

One of the recommendations from Phase 2 of the task order is to increase the fidelity of the NASA Ames hardware fidelity. Currently, NASA Ames has several MDMs enclosed in STARS consoles, but no DSR console frames, DSR keyboards or trackballs.

#### **1.1.13.2.1 Console Frame**

The production of DSR consoles ended in 1996, so procuring the R-position or D-position console frame is difficult. There are three possible options:

1. Request existing frame from the FAA at an approximate cost of \$15,000 per R-position or D-position console frame.
2. Manufacture new frames. This cost would be greater than purchasing existing frames from the FAA since many of the parts of the console frame would have to be manufactured from scratch.
3. Have a DSR Mockup console manufacture. This would cost approximately \$6,500 for each R and D-position console.

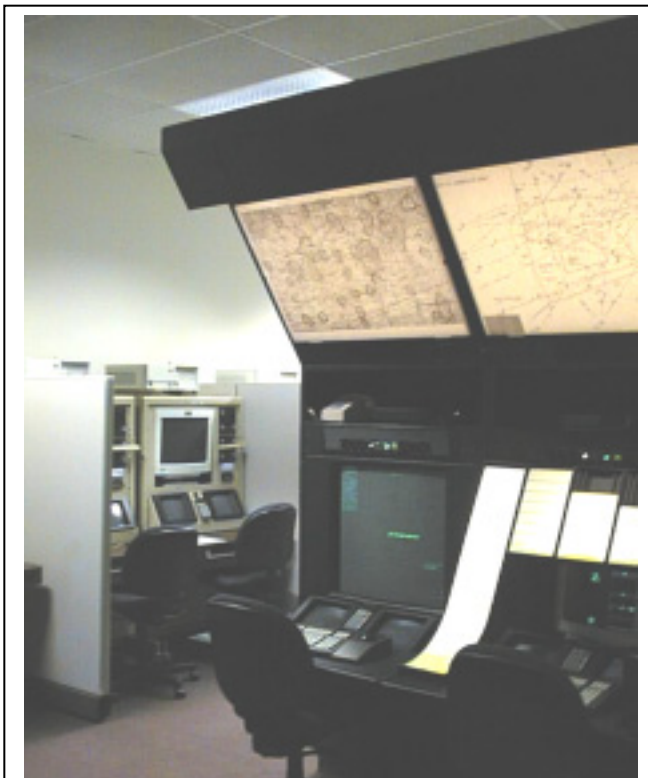


Figure ~~1414~~14 DSR Console Frame

### **3.2.1.1 Technical Issues**

None.

### **3.2.1.2 Summary of Advantages and Disadvantages**

The choice between the three options is a trade between cost and fidelity. At the high end of the fidelity is the remanufacture or purchase of existing consoles from the FAA. At the middle area is the manufacture of mockup consoles at a lower cost.

### **3.2.1.3 Recommendation**

As specified in ~~Table 3~~Table 3, it was recommended that NASA Ames upgrade their DSR Console fidelity. Currently NASA Ames has several Sony MDMs enclosed in STARS consoles. Since high hardware fidelity is not a requirement, it is recommended option is to request mockup console frames be manufactured. The exterior of the console will have the same look and feel, but parts internal to the console will be substituted. This solution will provide sufficient fidelity at a reasonable cost that will support the NASA Ames research and development needs.

## **3.2.2 Keyboard and Trackball**

Another way to increase the fidelity of the DSR system is to provide DSR keyboards and trackball for each console. The DSR trackball connects directly to the DSR keyboard and the DSR keyboard connects to the CDG via an RS-422 connection.

The recommended DSR simulator option for the NASA Ames simulator is the hybrid solution (refer to section 3.1.4). Since this is an X-Windows solution, the DSR keyboard is a client that attaches to the X-Server. Physically, the DSR keyboard connects to a serial port of a PC through an RS 232 connection.

The X-Server within the PC processes the keyboard commands. The application to process the keyboard data can either be embedded into the X-Server, or an extension, such as XtestExtents. Presently a keyboard driver has been written for the Sun Solaris operating system that makes use of the XtestExtents. In addition, this approach was used for the DSR D-position that attaches a DSR keyboard to an X-Server.

Cortron no longer manufactures the DSR keyboards and trackballs. In fact, only replicas can be built because certain parts in the keyboards are no longer available. In order to procure these, Cortron would need to be contracted to manufacture them.



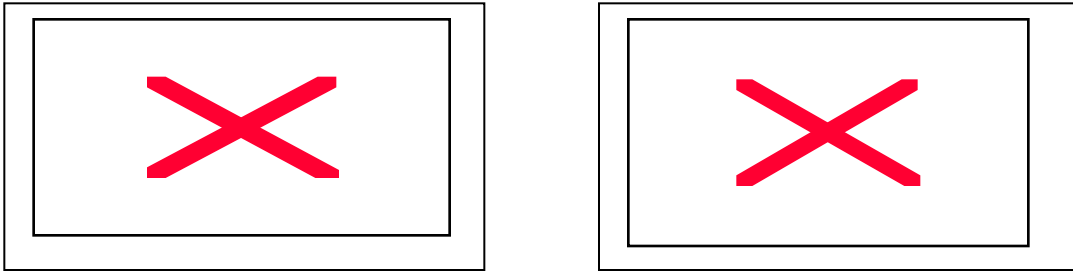


Figure 15-15 DSR Keyboard and Trackball

#### 3.2.2.1 *Technical Issues*

None

#### 3.2.2.2 *Summary of Advantages and Disadvantages*

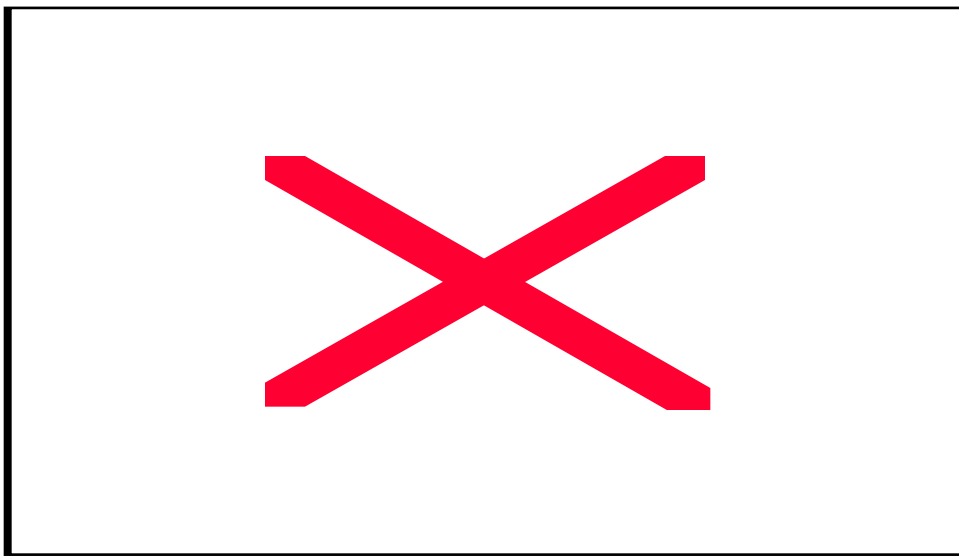
The DSR trackball will provide the controllers with the trackball that they normally use. By having the DSR keyboard, the controllers can enter the complete set of characters, such as the clear weather symbol, into the computer readout device. Also, the use of the Xtest Extension has already been written for the DSR simulator.

#### 3.2.2.3 *Recommendation*

The DSR keyboard and trackball should be purchased from Cortron and the keyboard driver obtained from LMATM. The Xtest extension should be used connect the keyboard to the X-Server.

#### 3.2.3 **Cost**

The approximate hardware cost for the consoles, trackballs, keyboards and flight strip



printers is illustrated in Table 8.

### 3.3 NASA Ames to DSR Simulator Interface

This section specifies the high level interface between the NASA Ames systems and the DSR simulator.

Currently, the AOL and the CTAS group require both simulation data from Host and to also display data at the DSR simulator display positions. Since the CTAS system currently does not receive data from the ADRS system, it makes sense that the DSR simulator exchange data with both the ADRS and the HADDs subsystems. The dual interface is shown in [Figure 16](#)[Figure 16](#)[Figure 16](#).

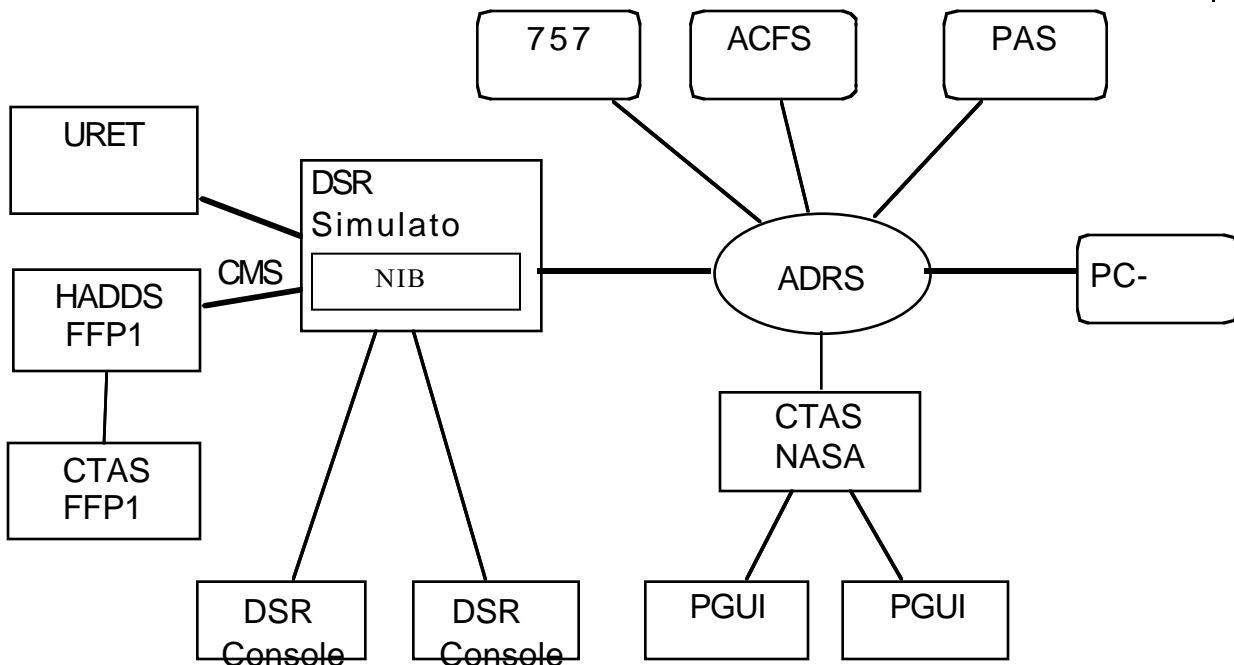


Figure [16](#)[16](#)[16](#) NASA Ames to DSR/Host Simulator

The ADRS is designed to handle multiple inhomogeneous interfaces and to add new interfaces in a fairly straightforward fashion. ADRS will be modified to integrate a new interface that is tailored to the DSR needs. This means the ADRS can function as server or client to multiple DSRs and use a specified handshake protocol, data structures, encoding and decoding that is specific to this particular interface.

The HADDs system is a Free Flight Phase 1 (FFP1) product rather than a simulator and is not expected to be easily modified. However, without modification, HADDs can only interface with the Host Substitute using the CMS.

---

## 4 Concluding Recommendations

This section is composed of three sections:

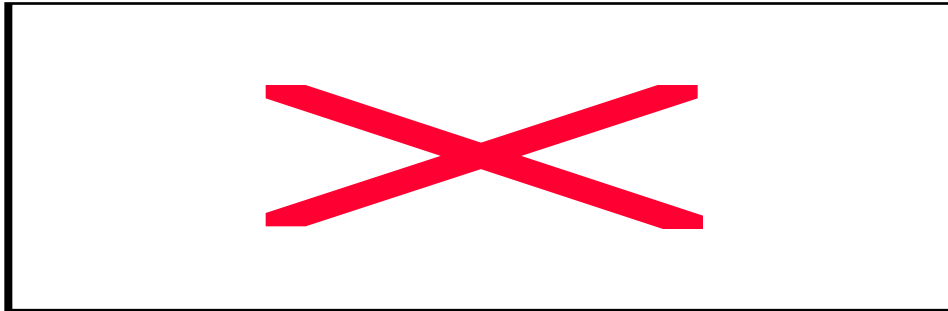
- Summary of the recommended hardware and software for DSR simulator.
- Proposed use of the system.
- Future work.

### 4.1 System Summary

This section summarizes the proposed composite solution for the NASA Ames DSR simulator. It has been recommended that:

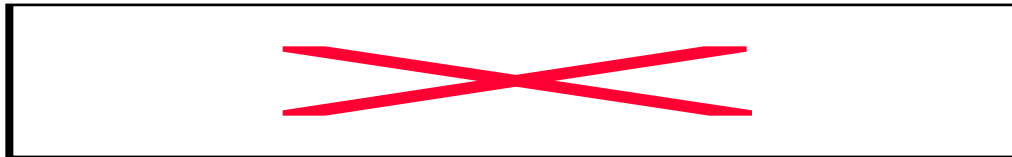
- The hybrid DSR Simulator solution be adopted as recommended in section 3.1.4. This solution provides simulated R and D-positions and a stimulated Host.
- Both R and D-position console mockup hardware be procured.
- For the R-position, a DSR trackball and keyboard be procured.
- For the D-position, a DSR keyboard, display and flight strip printer be procured.
- The current NASA Ames VSCS simulator system continues to be used for air to ground communications and voice recording.
- The DSR simulator should be physically attached to the existing Switched Ethernet LAN at NASA Ames, as directed by NASA Ames.
- Addition hardware is necessary to provide the DS Gateway used for the CTAS interface. This hardware cost is not included in the hardware estimate.

[Table 9](#)[Table 9](#)[Table 9](#) summarizes the estimated procurement cost, less installation, maintenance, DSR trackball and DSR keyboards. There is no cost for the stimulation option since costs are not available for the DSR hardware. Also, for the simulation and the hybrid solutions, the cost is separated by having only an R-position, D-position, or both R and D-positions.



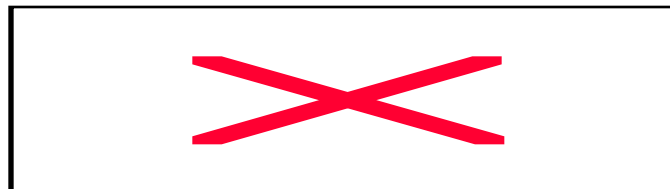
*Table 9 Total Hardware Procurement for Three Options*

This system is estimated to have the following approximate software development costs. The bulk of the development is in the completion of the R-position command interface to Host and integrating the R-position simulator on a Flight Deck Simulator Infrastructure. Notice that a range of SLOC has been provided since more study is necessary to size the effort. Also, this estimate does not include the SLOC necessary to implement new interfaces between DSR and the NASA Ames systems or to change the position software necessary for prototyping.



*Table 10 Software Development Cost Range Summary*

In addition, since DSR system is currently being upgraded by new functionality in release BCC21, this new functionality must also be integrated into the prototype. The majority of the functionality is associated with the R-position FDB. The new function and its approximate size is listed in ~~Table 11~~~~Table-11~~~~Table-11~~.



*Table ~~11~~~~11~~~~11~~ Future Software Development Work*

This large amount of development to complete the DSR simulator is offset by the benefits of having a system:

- Whose platform is not end of life (e.g., not RISC/6000, CDG or token ring based)
- That is easier to modify for prototype purposes because it is implemented in standard X and a high-level graphics language.

## 4.2 DSR Simulator Use

This section considers how an integrated DSR simulator might be used at NASA Ames. As seen in [Figure 11 Hybrid Functional Architecture](#), the inputs to the system are from HADDs, ADRS, SIM tape and from the User Interface. These inputs stimulate the system in the following manner:

- Host direct stimulation of DSR and ADRS.
- Controller stimulation of Host.
- ADRS stimulation of Host with radar data.
- ADRS stimulation of DSR with new message.
- Controller stimulation of ADRS with new message.
- HADDs Host interaction.

The stimulation types and the resultant system behaviors are illustrated in the sections that follow.

### Host Direct Stimulation of DSR and ADRS

A SIM tape is loaded on the 3480 tape drive ([Figure 17](#)). The Host substitute is commanded to read the simulation tape, stores it on hard disk, and then executes the scenario. The Host substitute forwards the data to the DIU (a), which distributes the radar and flight data to the ADRS and R and D-positions for display.

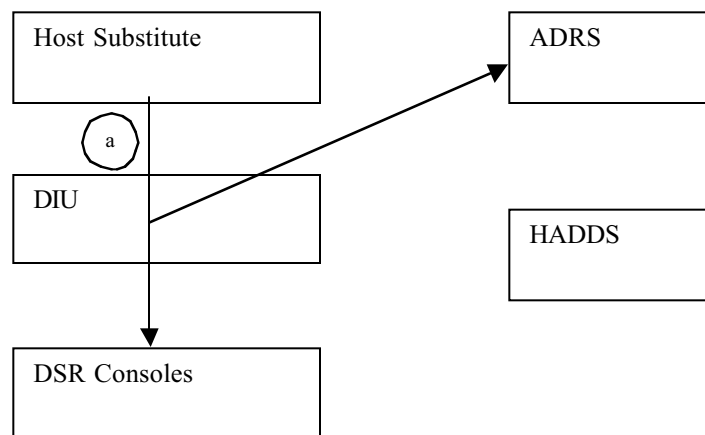
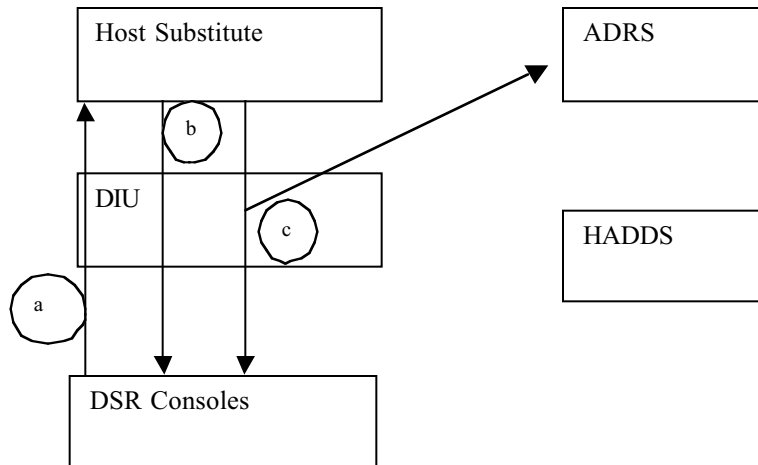


Figure 17 Host Direct Stimulation of DSR and ADRS

## Controller Stimulation of Host

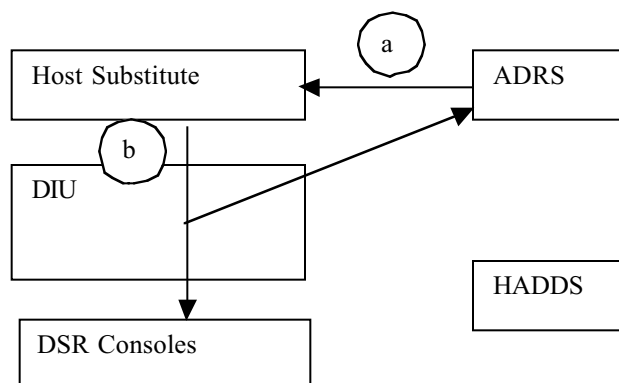
~~Figure 18~~~~Figure 18~~~~Figure 18~~ illustrates controller commands that are entered at the R or D-position console (a). The message is transmitted to the DIU. The DIU routes the message to the Host substitute and the Host substitute responds to the controller (b). If the command results in the output of data by Host, such as a flight plan update, then flight data is output by the Host substitute to the DIU (c). The DIU distributes the data to all of its clients (ADRS and the DSR consoles).



*Figure ~~18~~~~18~~~~18~~ Controller Stimulation of Host*

## ADRS Stimulation of Host with Radar Data

~~Figure 19~~~~Figure 19~~~~Figure 19~~ illustrates ADRS transmitting radar data in common digitizer format directly to the Host Substitute (a). The PAMRI within the Host Substitute converts the data and transmits the data to the Host via the Host channels for processing.



Host outputs the radar data (b) to the DIU and the DIU routes the data to the clients (c).

Figure ~~191919~~ ADRS Stimulation of Host with Radar Data

### **ADRS Stimulation of DSR with New Message**

~~Figure 20~~~~Figure 20~~~~Figure 20~~ illustrates ADRS transmitting a new message to the DSR console via the DIU (a). The DIU routes the message to the proper sector position. The DSR position simulator has been modified to receive the new message, update any databases, and display the new message data using X protocol. If the data is being displayed on the R-position, then the data must be integrated with the current DSR design (e.g. pick and draw precedence). The DSR position provides status back to ADRS (b).

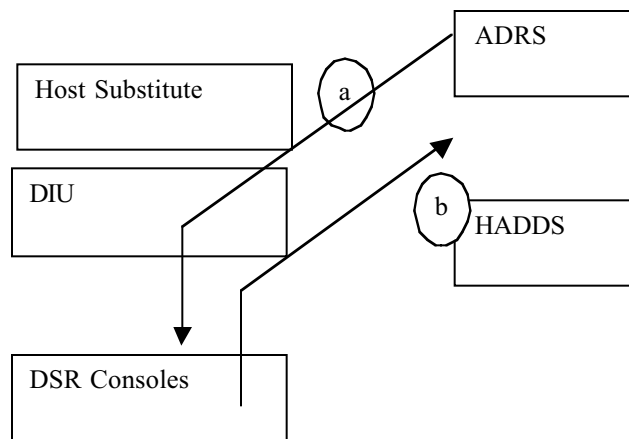


Figure ~~202020~~ ADRS Stimulation of DSR with New Message

### Controller Stimulation of ADRS with New Message

Figure 21 illustrates a controller command being entered at a controller position and being transmitted to the DIU (a). The DIU routes the non-DSR message to the ADRS system (a) and receives status (b). Then the ADRS system sends data for display at the controller position (c). If the data is being displayed on the R-position, then the data must be integrated with the current DSR design (e.g. pick and draw precedence).

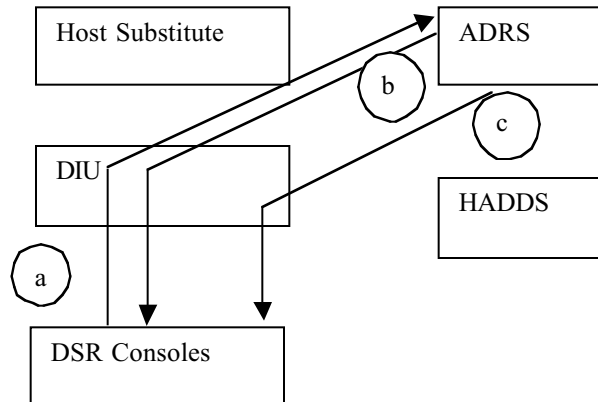


Figure 21 Controller Stimulation of ADRS / HADDS with New Message

### HADDS Host Interaction

Figure 22 illustrates CMS data being transmitted between the Host and the HADDS. In this scenario, the HADDS has transmitted metering data to the Host (a) that is output by the Host to the DIU (b). The DIU sends the data to its client's (c).

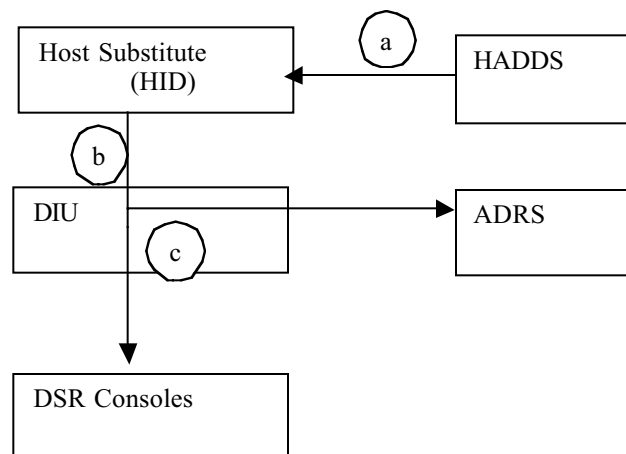


Figure 22 HADDS Host Interaction

### **4.3 Future Work**

It is recommended that the next steps in the definition of this system be:

- Further investigate how NASA plans to use the DSR simulator. A higher fidelity is necessary if the system is used for controller evaluation, while a lower fidelity is necessary while running research scenarios. By further specifying the simulators use, the cost can be better estimated.
- Document the interface between DSR simulator and the NASA Ames subsystems in an Interface Control Document (ICD). The ICD would document the message interface between ADRS/HADDS and the DIU. The set of messages contained in the ICD would include Common Digitizer messages, interfacility flight messages, CMS, and NASA Ames unique display and command messages.
- Follow the DSR Standalone Trainer System to ensure that the NAS In a Box product stays suitable for use by the DSR simulator.
- Further define the mockup console frame by identifying the replacement parts. Once these parts are defined, ensure that it continues to be fit for use.
- Investigate recording and playback of controller actions using X Extensions. It would be very valuable for a research and development site be able to record and then playback the controller actions.
- Investigate SAR and playback of messages. This is a much higher level of recording than the X recording since for the SAR, messages are recorded rather than the controller actions (e.g. movement of the trackball).
- Investigate the display of NEXRAD weather data on displays. This was a desired function to increase the fidelity of the DSR simulator but was not investigated.
- Investigate VSCS system upgrade. Currently NASA Ames has an air-ground VSCS simulator, but no ground-to-ground system. Upgrades to this system could be considered. MARC contracted directly with Harris for a one of a kind system that provided a video display monitor that interfaced with a telephone switch. The touch screen controlled communications between the controller and the pilot. However, this solution is no longer available since the video display monitor is no longer manufactured.
- Investigate software version control for the DSR simulation software to allow rapid selection and loading of versions of the DSR simulator that have a particular prototype modification. This would facilitate multiple users to tailor the DSR simulator for their own use.

- Investigate facility fidelity, such as console seating and lighting.
- Investigate set of adaptation data needed by the DSR simulator.
- Investigate the implementation of URET functionality for the DSR simulator.

---

## 10 Appendix A: Host S/390 Emulation

For both the stimulation and the hybrid options, an S/390 emulation of actual Host code is recommended. Thus, since this section is shared between two of the options, this section has been placed in the appendix.

The method of emulating Host described in this section is dependent on the resultant solution for the DSR Standalone Training System (Task Order 98) program. This FAA task order has directed LMATM to perform an engineering analysis to determine options for providing a stand-alone training capability for use with the current DSR Test and Training Systems found at the ARTCC centers. The system is considered ‘standalone from Host’ because the solution does not depend on the operational Host. Instead, the host that drives the training system is found in a separate processor: NAS In A Box.

The outcome of this task order will be a recommended architecture and hardware for a subsystem that will control and execute Host programs. This program is planned for deployment possibly in 2002. Accordingly, this is a timely solution that may be utilized by the NASA Ames DSR simulator.

### **1.110.1 Overview**

The FLEX-ES™ product from Fundamental Software, Inc. emulates the ESA/390™ (System/390 ©) mainframe computer architecture on an Intel® Pentium® processor. This software emulation of IBM System/390 hardware creates a platform for running mainframe operating systems and/or applications without modification to the operating system or Host application. The FLEX-ES product runs as an application program under the SCO-UNIX® operating system.

FLEX-ES provides support for IBM peripherals in the following ways:

- As emulated devices where the FLEX-ES software emulates supported hardware devices.
- As channel-attached peripherals through the FLEX-ES Parallel Channel Adapter (PCA) that provides Bus and Tag connectivity. The PCA provides 3 channels per card and supports most peripherals with transfer rates of both 3.0 and 4.5 megabytes (Mb)/second in streaming mode.
- As Small Computer System Interface-attached physical devices.

This Host substitute solution requires a single Pentium based FLEX-ES system. The DSR Standalone Training System uses Virtual Machine and NASSERV software to run multiple Host training sessions at each ARTCC site. However, the NASA Ames DSR simulator only needs to run a single Host and thus does not require NASSERV software.

It is recommended that the Virtual Machine license be procured to allow Host to have virtual channels rather than hard addressed channel addresses. The benefit of virtual channels is that multiple processes can use a single channel port. This reduces channel contention.

### **1.210.2 System Architecture**

The FLEX-ES system is capable of running SCO UNIX, 390 Emulation, and Host. Peripheral devices such as tape drives and disks are connected using a standard SCSI connection. This system uses SCO UNIX as the base Operating System (OS) and runs Fundamental Software's 390 Emulation Software to simulate an IBM System/390. This software emulation of IBM System/390 hardware creates a platform for running mainframe operating systems and/or applications without modification to the operating system or application (i.e. Host software).

Connection to the DSR positions and NASA Ames subsystems is provided through channel interfaces to the Sunhillo Corporation Host System Interface (HSI) machine (refer to [Figure 23](#)~~Figure 23~~[Figure 23](#)). All data flows through the HSI. The HSI machine will use UNIX.

Radar and flight simulation data typically will flow from the ADRS over the Ethernet LAN, through the HSI over the channel interface, and to the Host session. The PCI to Channel Adapter (PCCA) is a PCI bus i960 co-processor card that contains drivers that interface to an IBM 390/370 channel. The PCCAs are found in the HSI while the IBM 390/370 channels are found in the Host substitute.

Data destined for the flight strip printer and DSR positions flow from a single Host session to the HSI through the appropriate channel port and PCI slot. The HSI routes DSR data over the Ethernet LAN to the DIU. The DIU act as a gateway between the Host processor channel interface and token ring LANs. Host Computer Interface Software (HCIS) resides in the DIU and provides support for the data transfer between the attached Host processing and the DSR applications. After decoding the messages, HCIS forwards these messages to the appropriate R-position and D-position via the LCN network. HCIS runs on a RISC/6000.

The HSI also routes flight strip data to the PCU emulation software where it is forwarded to the appropriate Flight Strip Printer using an EIA RS-232 or RS-422 interface.

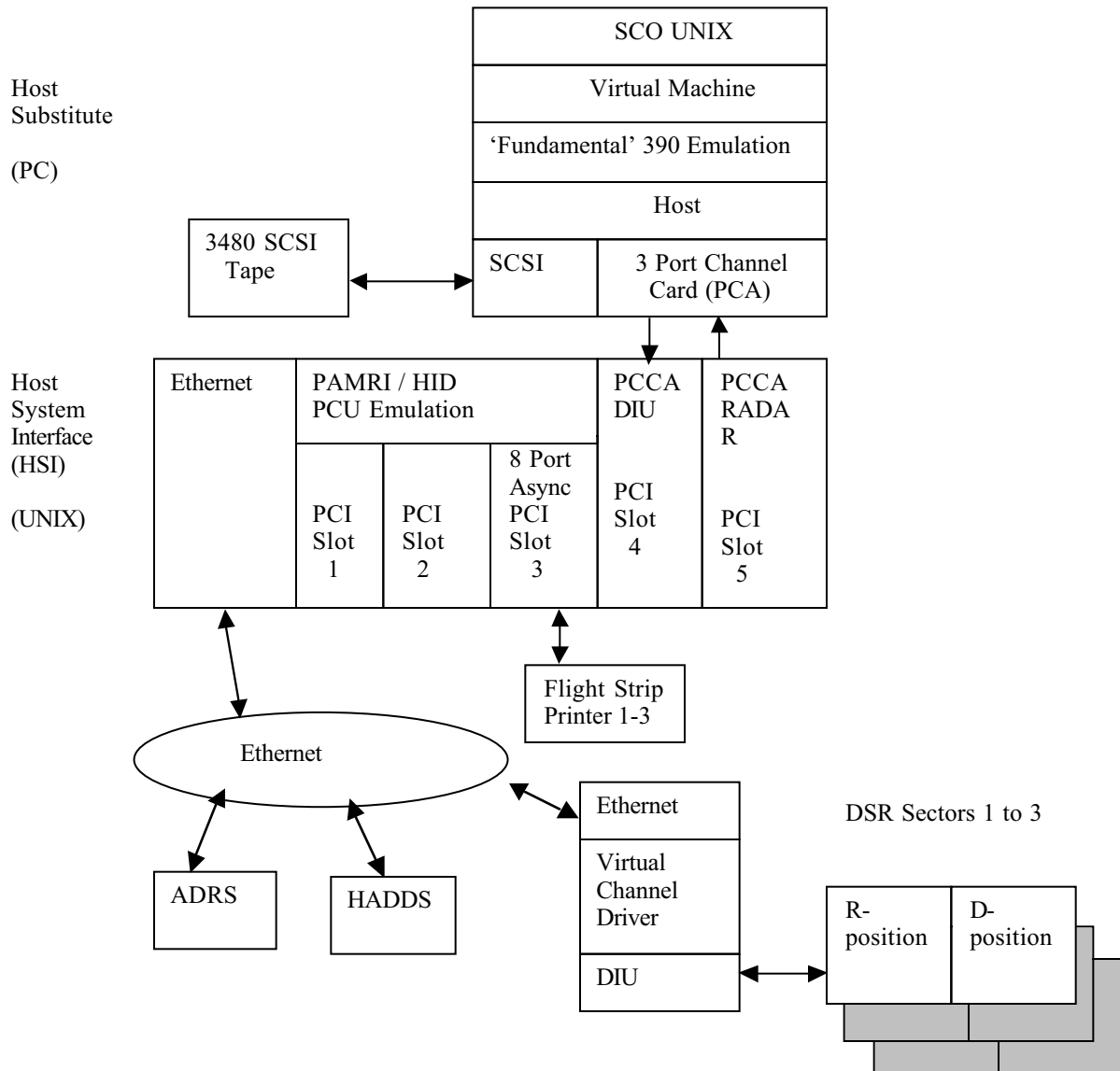


Figure 232323 Host S/390 Emulator with Peripherals

The Host system is loaded from a tape read from the 3480 tape drive through the SCSI interface. A file call the Disk Dump Data Restore (DDR) is read from the 3480 tape and is stored on a virtual disk, available to Host to read. The DDR file may be procured from the FAA. In order to start Host, the user informs Host of the location of the DDR data

from the PC terminal. The user specifies the DDR location in the ‘Deck File’ and sends the Deck File to Host. The Deck File also points Host to the location of the Host adaptation data. The Host executable is loaded into memory and the S/390 emulator processes the commands found in the Deck File.

Scenario data that drives the Host system is recorded on a virtual SIM tape. SIM tapes can be generated by hand. To start a scenario, a ‘SIM’ tape is loaded on the 3480 tape unit. This file is copied onto the hard disk as a virtual ‘SIM’ tape. This is read by Host and is processed by Host.

Long range radar data from ADRS is transmitted to the PARMi emulator in the HSI, through the Ethernet using Common Digitizer 2 format. PARMi converts the data format into a format that is expected by Host (refer to [Figure 24](#)[Figure 24](#)[Figure 24](#)).

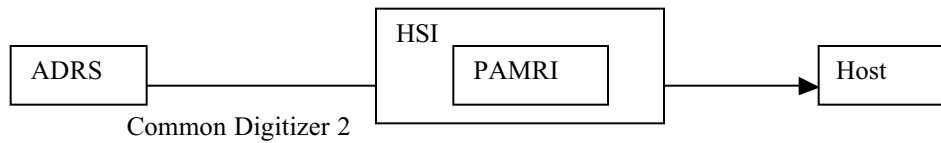


Figure [24](#)[24](#)[24](#) Host Radar Input Logical Thread

### 10.3 Technical Issues

Differences between requirements of the DSR Standalone Training System and the NASA Ames DSR simulator are summarized in the table below:

Function	DSR Standalone Training System	NASA Ames DSR Simulator
Token Ring	The DIU is designed to connect to a token ring.	If the hybrid solution is selected, then the DIU must be modified to connect to an Ethernet.
Host Sessions	Since it is a training system, multiple Host Sessions are required. Thus NASSERV is required.	Only a single Host Session is required, so NASSERV is not required.

Table [12](#)[12](#)[12](#) DSR Standalone Training System and NASA Ames DSR Simulator Differences

This solution has a dependency on the successful completion of the DSR Standalone Training System, due for completion in the year 2002. The design of this system is not yet complete and there are still technical issues to resolve.

In order to upgrade this system to a multi-Host capability, no additional hardware is necessary since multiple copies of the Host software can be run on the processor. Additional control software (NASSERV) will be necessary to control the multiple hosts however.

## 10.4 Cost

The table below illustrates the cost of a Host Stimulator system. This cost is incurred by the stimulator and hybrid solutions.

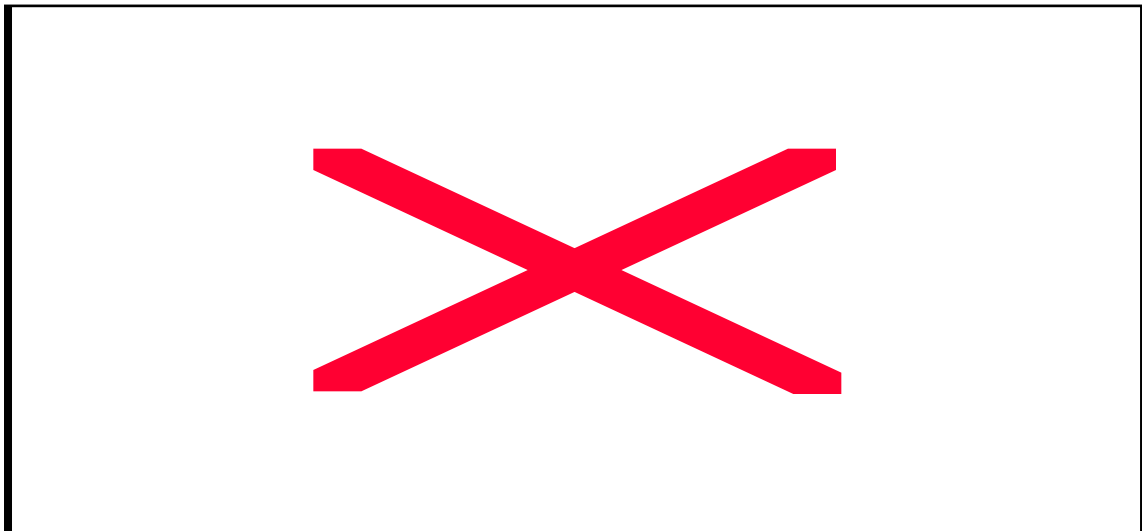


Table ~~131313~~ Host Stimulator Product Cost

It is recommended that the Virtual Machine license be purchased. Note that the fee for Virtual Machines is a single \$20,000 fee. The fee increases and becomes a monthly fee when the Host processor is run above 8 MIPS. However, even with 200 aircraft running Direct Sim on the Host stimulator, the processor is thought to run below 8 MIPS.

---

## 20 Appendix B: DSR Simulation

Per FAA task order, LMATM has developed three display application prototypes to a DSR baseline specification subset by three independent development teams. The goal of project was to evaluate the probable replacement of the DCX and Remote Graphics Language by graphic accelerator cards (TechSource and Barco graphic cards) and X Windows.

The set of X Windows products that were evaluated were:

- Eagan (ViewMan)
- Gallium (InterMAPhics)
- Orthogon (ODS Toolbox)

Each team developed its own application prototype from a subset of the DSR baseline specifications for the R-position software. All the teams gathered at the Integration and Interoperability Facility at the William J. Hughes Technical Center for final integration, test and demonstration of their prototype to a team of controllers. The goal of each team was to develop a prototype that looked exactly like the DSR R-position.

The applicable study results were:

- All products underlying the application prototypes are useable.
- Application prototypes built on any of the underlying products are maintainable.
- All application prototypes passed the functional tests and were stable throughout.
- All application prototypes met the performance and storage requirements.
- The Eagan and Gallium prototypes demonstrated significantly better CPU utilization performance than the Orthogon prototype.
- The Eagan and Gallium prototypes demonstrated good response times and small track display latency times compared with the Orthogon prototype.
- The Eagan prototype demonstrated the most consistent and smoothest response times.
- The coding efforts of each of the three candidates appear to be very close.
- The coding efforts of the candidates are not much better than for the DSR baseline.

- One application prototype cannot easily be ported from one product to another.
- Solutions were not sensitive to which graphics card was used.
- Solutions were not sensitive to Sony MDM versus Barco ISIS flat panel.
- X Windows (versus Remote Graphics Language) provides an improved development environment.

Based on these results, the generation of a DSR simulator from one of these products seems feasible. The section below lists the DSR functionality that has already been coded into these prototypes.

### **1.120.1 R-position Console Simulation**

Currently, there is no complete DSR simulator available. However, there is a DSR simulator that has been initiated and currently provides the following the set of simulated DSR functions:

<b>Function</b>	<b>Applicable View</b>	<b>Description</b>
Annotations	Situation Display	Most of the annotation functions are provided. Annotation function added a toolbar and provided graphics functions to the Situation Display view.
General Drawing Characteristics	All	Draw all items so that they have the same appearance as their baseline DSR Human Machine Interface counterparts. Includes fonts, line styles, fill patterns, color schemes, etc. but does not include anti-aliasing.
Drawing Precedence and Transparency	All	Draw all items opaque / transparent and above or below other types of data as appropriate.
Basic View Manipulation	All	Display views, Move, Expand / Collapse views as in DSR.
Basic Trackball Capabilities	All	Trackball cursor and button actions including wrapping, cursor size, shape, and placement in conjunction with key or button actions.
Keyboard Inputs	R-CRD	Keystroke echoing, key actions (cursor movement, etc.).
Full Data Blocks	Situation Display	Three-line full data block content, leader lines to northeast of track symbol.
Basic Command Composition	R-CRD	Message Composition Area, Keyboard inputs, Trackball Pick processing (partial).
Basic Command Feedback	R-CRD	Feedback Area and Response Area.

<b>Function</b>	<b>Applicable View</b>	<b>Description</b>
Basic DC View Capabilities	Display Controls and Status View	Display the Display Control View, including panels and buttons, panel display/removal management, button highlighting.
DC Command Capabilities	Display Controls and Status View	Integrated with trackball actions and Situation Display. Includes Filtering, range, vectors, histories, brightness controls, cursor size, leader line lengths, cursor speed.
Geomap Data	Situation Display	Drawing map data. Does not include anti-aliasing. Includes correct drawing precedence and transparency.
Data Block Offset	Situation Display	Full data block and leader line offset to the 8 cardinal directions from the track symbol.
Leader Line Length	Situation Display	4 leader line lengths (including zero length).
Limited Data Blocks	Situation Display	<ul style="list-style-type: none"> <li>• All DSR limited data blocks function.</li> <li>• Limited data blocks produce a significant load on the DSR system.</li> </ul>
Pick Precedence	All	Implement trackball pick precedence rules as in the DSR Human Machine Interface.
Range Data Blocks	Situation Display	<ul style="list-style-type: none"> <li>• A subset of range data blocks functions was required. A limited syntax to create and control the range data blocks was implemented.</li> <li>• Range data blocks added data to the track data block and added a new view (range data block list).</li> </ul>
Situation Display Insets	Situation Display	<ul style="list-style-type: none"> <li>• A simple inset implementation was required. This inset was suppressible, movable, was opaque to data displayed on the situation display, and had pick and draw precedence. The inset was not resizable.</li> </ul>
Target Data	Situation Display	Draw with correct symbols and placement.
Target Histories and Aging	Situation Display	Draw with correct number of histories, aging.
Track Data	Situation Display	Draw with correct symbols and placement.
Track Filtering	Situation Display	Integrated with Display Control view.

Function	Applicable View	Description
Track Dwell	Situation Display	<ul style="list-style-type: none"> <li>Track data block dwell, not user dwell.</li> <li>Dwell added features to the track data block.</li> </ul>

Table ~~141414~~ Existing DSR Simulator Functions

In order to have a useful DSR simulation, at a minimum the following DSR functions must be completed.

Function	Applicable View	Description
Commands to Host	Computer Readout Display	The transmission of a set of commands to Host. Currently no commands can be sent to Host. This includes parsing of the Host commands.
System Infrastructure	none	The current DSR R-position does not have any infrastructure functions such as system time and message transmission using TCP/IP. It is recommended that the R-position be integrated into a FlightDeck simulator platform.

Table ~~151515~~ Required Upgrades to DSR Simulator

The following DSR functions listed in ~~Table 16Table-16Table-16~~ are not complete and have not been included in the software development estimate. It is felt that the simulation priority of these functions is not as high as the functions listed in ~~Table 15Table-15Table-15~~.

Function	Applicable View	Description
EDARC Capabilities	EDARC Situation Display	The EDARC system provides a backup situation display for the DSR system.
Flight Plan View	Situation Display	The R-position can request flight plan readouts.
Longitudinal Scale	Host Situation Display`	A longitudinal scale is displayed on the situation display at a user specified position, orientation and length.
Non-Interactive See-All	Computer Readout Display	An authorized R-position can request an exact copy of a target sector's Host Situation Display.

<b>Function</b>	<b>Applicable View</b>	<b>Description</b>
Quick Action Keys	R-CRD	Clear, Acknowledge, command verb keys on the keyboard provide quick actions for the controller.
Quicklook	Situation Display	An R-position can request the display of FDBs from up to five other sectors.
Reconstitution	All	This process resynchronizes the DSR system with the Host. DSR requests data from Host to repopulate its state data.
VSS Sensing	R-CRD	Sensing of the Video/Serial Switch (VSS) position that is used when switching between EDARC and Host situation displays.

Function	Applicable View	Description
User Preferences	All	User preferences can be saved and restored.

Table ~~161616~~ Excluded Functions from DSR Simulator

## 20.2 D-position Console Simulation

This study has not addressed URET CCLD, but is recommended for future study. According, only the D-CRD, Time, and Tool Bar views will be displayed at the simulated D-position.

In order to have a useful DSR simulation, at a minimum the following DSR functions must be completed.

Function	Applicable View	Description
Commands to Host	Computer Readout Display	The transmission of a set of commands to Host. This includes parsing of the Host commands.
Time	Time View	This view displays system time.
Tool Bar	Tool Bar	The tool bar view allows the display of the flight plan view.

Table ~~171717~~ D-position Simulation

---

## 30. Appendix C: Acronyms

<b>ACFS</b>	Advanced Concepts Flight Simulator
<b>ADAR</b>	ARTS Data Acquisition and Routing
<b>ADRS</b>	Aeronautical Datalink and Radar Simulator
<b>API</b>	Application Program Interface
<b>ARTCC</b>	Air Route Traffic Control Center
<b>CAASD</b>	Center for Advanced Aviation System Development (Mitre)
<b>COTS</b>	Comercial Off the Shelf
<b>CM</b>	Communications Manager
<b>CTAS</b>	Center TRACON Automation System
<b>DIU</b>	DSR Interface Unit
<b>DDR</b>	Disk Dump Restore file
<b>DSR</b>	Display System Replacement
<b>DSSC</b>	DSR System Support Complex
<b>ERAU</b>	Embry Riddle Aeronautical University
<b>FSP</b>	Flight Strip Printer
<b>HADDS</b>	Host ATM Data Distribution System
<b>HCIS</b>	Host Computer Interface System
<b>HFL</b>	Human Factors Lab (at William J. Hughes Technical Center)
<b>HIS</b>	Host System Interface
<b>HDAR</b>	Host Data Acquisition and Routing
<b>ICA</b>	Integrated Communication Adapter
<b>ISM</b>	Input Source Manager
<b>I2F</b>	Integration and Interoperability Facility (at William J. Hughes Technical Center)
<b>LAN</b>	Local Area Network
<b>LCN</b>	LAN Computer Network
<b>LGSM</b>	Local/Group System Monitor and Mode Management
<b>LMATM</b>	Lockheed Martin Air Traffic Management
<b>MARC</b>	Mid-America Aviation Resource Consortium
<b>MPI</b>	Multiple Purpose Interface
<b>PAMRI</b>	Peripheral Adapter Module Replacement Item
<b>PAS</b>	Pseudo Aircraft Simulator
<b>PCA</b>	Parallel Channel Adapter
<b>PCI</b>	Peripheral Component Interconnect
<b>PCU</b>	Printer Control Unit
<b>RISC</b>	Reduced Instruction Set Computer
<b>SAR</b>	System Analysis Recording
<b>SCO</b>	Santa Cruz Operation

<b>SCSI</b>	Small Computer System Interface
<b>STA</b>	Scheduled Time of Arrival
<b>TAP</b>	Terminal Area Productivity
<b>TIL</b>	Technology Insertion Lab (Lockheed Martin)
<b>UTP</b>	Unshielded Twisted Pair
<b>VM</b>	Virtual Machine
<b>WARP</b>	Weather and Radar Processor
<b>WDAD</b>	Weather Data Acquisition Daemon
<b>WDPD</b>	Weather Data Processing Daemon